# Table of Contents

1. Cover Page .................................................. 1
2. Table of Contents ........................................... 2-5
3. Welcome ....................................................... 6
4. Installation and Maintenance ............................... 7
5. Security Overview .......................................... 8
   5.1. Rights Based Security .................................. 8-11
   5.2. Role Based Security .................................... 11-13
6. SentryOne Client ............................................. 14
   6.1. Client Interface ......................................... 15-16
      6.1.1. Navigator Pane ..................................... 16-18
      6.1.2. Event View Pane .................................. 18-19
      6.1.3. Conditions Pane ................................. 19-21
      6.1.4. Settings Pane ..................................... 21-25
      6.1.5. Client Alerts ..................................... 25-26
      6.1.6. Menus ............................................... 26-29
      6.1.7. User Preferences .................................. 29-33
      6.1.8. License Management ............................... 33-35
   6.2. Navigator Nodes ........................................ 35
      6.2.1. Favorite Targets ................................... 35-36
      6.2.2. Virtualization ..................................... 36
      6.2.3. All Targets ......................................... 36-38
         6.2.3.1. Default Site .................................. 38-39
         6.2.3.2. Targets ......................................... 39-41
         6.2.3.3. Instances ....................................... 41-42
      6.2.4. SQL Server Registrations .......................... 42
      6.2.5. Contact Management ................................ 42-45
      6.2.6. Schedules and Windows ............................ 45-48
      6.2.7. Object Groups ...................................... 48-51
      6.2.8. Monitoring Service ................................. 51
         6.2.8.1. Monitoring Service Settings .................. 51-55
            6.2.8.1.1. SMTP Configuration ......................... 55-58
         6.2.8.2. Site Configuration ............................. 58-60
         6.2.8.3. Inventory View .................................. 60-61
         6.2.8.4. Monitoring Service Load Balancing and Fault Tolerance 61
         6.2.8.5. Actions Log ..................................... 61-63
         6.2.8.6. System Status ................................... 63
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.8.7</td>
<td>Service Configuration Utility</td>
<td>63-64</td>
</tr>
<tr>
<td>6.2.8.8</td>
<td>Monitoring Service Logon Account</td>
<td>64</td>
</tr>
<tr>
<td>6.3</td>
<td>Common Tasks</td>
<td>64</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Reporting</td>
<td>64-71</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Adding Notes</td>
<td>71-73</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Monitor Additional Instances</td>
<td>73-75</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Connect to Installation</td>
<td>75-76</td>
</tr>
<tr>
<td>7</td>
<td>Alerting and Response System</td>
<td>77-78</td>
</tr>
<tr>
<td>7.1</td>
<td>Conditions</td>
<td>78-80</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Condition Settings</td>
<td>80-84</td>
</tr>
<tr>
<td>7.1.1.1</td>
<td>Filter Editor</td>
<td>84-85</td>
</tr>
<tr>
<td>7.1.2</td>
<td>General Conditions</td>
<td>85-92</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Failsafe Conditions</td>
<td>92-94</td>
</tr>
<tr>
<td>7.1.4</td>
<td>Audit Conditions</td>
<td>94-96</td>
</tr>
<tr>
<td>7.1.5</td>
<td>Advisory Conditions</td>
<td>96-102</td>
</tr>
<tr>
<td>7.2</td>
<td>Actions</td>
<td>103-109</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Action Behaviors</td>
<td>109-111</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Message Editing</td>
<td>111-117</td>
</tr>
<tr>
<td>7.3</td>
<td>Settings</td>
<td>117-120</td>
</tr>
<tr>
<td>7.3.1</td>
<td>SQL Server Settings</td>
<td>120-126</td>
</tr>
<tr>
<td>7.3.2</td>
<td>SSAS Settings</td>
<td>126-127</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Windows Settings</td>
<td>127-129</td>
</tr>
<tr>
<td>7.4</td>
<td>Response RuleSets</td>
<td>129-132</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Response RuleSet Examples</td>
<td>132-136</td>
</tr>
<tr>
<td>7.5</td>
<td>Output Content Match</td>
<td>136-137</td>
</tr>
<tr>
<td>8</td>
<td>SentryOne Event Manager</td>
<td>138</td>
</tr>
<tr>
<td>8.1</td>
<td>History Filter</td>
<td>138-139</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Filter Editor</td>
<td>139-140</td>
</tr>
<tr>
<td>8.2</td>
<td>Calendar View</td>
<td>140-144</td>
</tr>
<tr>
<td>8.3</td>
<td>List View</td>
<td>144-145</td>
</tr>
<tr>
<td>8.4</td>
<td>Custom Event Views</td>
<td>145</td>
</tr>
<tr>
<td>8.4.1</td>
<td>Working With Custom Event Views</td>
<td>145-148</td>
</tr>
<tr>
<td>8.4.2</td>
<td>Filtering Process</td>
<td>148</td>
</tr>
<tr>
<td>8.4.3</td>
<td>Filter Process Flowchart</td>
<td>148-149</td>
</tr>
<tr>
<td>8.4.4</td>
<td>Leveling Backups Across a SAN or NAS</td>
<td>149-151</td>
</tr>
<tr>
<td>8.5</td>
<td>Chaining and Queuing</td>
<td>151</td>
</tr>
<tr>
<td>8.5.1</td>
<td>Event Chains</td>
<td>151-160</td>
</tr>
</tbody>
</table>
8.5.2. Job Queuing
8.6. Clearing Failed Instances
8.7. Copy Job/Alert Wizard
8.8. Conflict Viewer
8.9. Event Object Information Window
8.10. Multi-server Administration (MSX/TSX)
8.11. Runtime Stats
8.12. Toolbars
8.13. Extended Job Step Logging / SSIS Logging
  8.13.2. Enabling SSIS Logging
8.14. Multi-server Administration (MSX/TSX)
9. SentryOne Performance Analysis
  9.1. Dashboard
    9.1.1. Dashboard Resolutions/Retention
    9.1.2. Baselining
    9.1.3. Performance Metrics
  9.2. Processes
  9.3. Disk Activity
  9.4. Disk Space
  9.5. Top SQL
    9.5.1. Plan Explorer (integrated)
      9.5.1.1. Plan Explorer Sessions
      9.5.1.2. SQLPerformance.com Plan Upload
      9.5.1.3. Plan Diagram Options
      9.5.1.4. Index Analysis
    9.5.2. Top SQL Runtime Stats
  9.6. AlwaysOn
  9.7. Query Plans
  9.8. Indexes tab
    9.8.1. Fragmentation Manager
  9.9. SSAS Usage Totals
  9.10. Block SQL
  9.11. Deadlocks
  9.12. Quick Traces
  9.13. Performance Analysis Overview
  9.15. Data Grids
Thank you for installing SentryOne. To get up and running in no time, please review the SentryOne Quick Start Guide.

If you have any technical questions, or for help with installation or configuration issues, please don’t hesitate to contact us:

**Email:** support@sentryone.com

**Phone:** 704-895-6241

**Toll Free:** 855-775-7733

**Support Forum:** [http://support.sentryone.com](http://support.sentryone.com)
Installation and Maintenance

**INSTALLATION**

Please refer to the [SentryOne Quick Start Guide](#) for detailed instructions on installation and setup of the [SentryOne Client](#) and the [SentryOne Monitoring Service](#).

**MAINTENANCE**

Just as with any other SQL Server database, it is important that regular maintenance activities be performed on the SentryOne database to ensure optimal performance. Please see the [SentryOne Database Maintenance](#) topic for more details and recommendations.
The **Quick Start Guide** covers the following topics related to SentryOne Security, including required permissions for the various SentryOne components.

<table>
<thead>
<tr>
<th>Security Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monitoring Service Security</strong></td>
<td>This topic discusses the permissions required by the <strong>SentryOne Monitoring Service</strong> account when watching (monitoring) Instances.</td>
</tr>
<tr>
<td><strong>Client Security</strong></td>
<td>This topic discusses the permissions required when running the <strong>SentryOne Client</strong>, including scenarios in which the Client connects directly to a monitored server.</td>
</tr>
<tr>
<td><strong>Watching Servers Across Domains</strong></td>
<td>This topic is a brief overview of the options available for watching (monitoring) servers across domains, including information about <strong>pass-through authentication</strong> and configuring <strong>SentryOne Sites</strong> within your environment.</td>
</tr>
<tr>
<td><strong>Non-Windows Environment</strong></td>
<td>This topic discusses the options for watching (monitoring) Instances in a non-Windows environment, including <strong>pass-through authentication</strong>.</td>
</tr>
<tr>
<td><strong>SentryOne Performance Analysis</strong></td>
<td>See this section for advanced information about the <strong>SentryOne Security Requirements</strong>, including <strong>Port Requirements</strong> for monitored servers.</td>
</tr>
<tr>
<td><strong>Azure SQL Database and SQL Data Warehouse</strong></td>
<td>This topic covers security aspects specific to Azure SQL Database and SQL Data Warehouse.</td>
</tr>
</tbody>
</table>

The **User Guide** covers the following topics related to restricting user access within SentryOne.

<table>
<thead>
<tr>
<th>Security Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rights Based Security</strong></td>
<td>This topic discusses restricting user access within the <strong>SentryOne Client</strong> based on Windows and SQL Server Authentication accounts.</td>
</tr>
<tr>
<td><strong>Role Based Security</strong></td>
<td>This topic discusses restricting user access within the <strong>SentryOne Client</strong> based on <strong>SentryOne Database</strong> roles.</td>
</tr>
</tbody>
</table>

5.1 **Rights Based Security**

SentryOne supports restricting server visibility within the SentryOne Client through the application of **Rights Based Security**. Both users and groups can be assigned a limited set of visible Sites, Target Groups, or Instances effectively restricting what the logged-in user can see. Certain objects and commands are also disabled for restricted users, see **Objects Hidden from Restricted**
Users for more information. Configuring Rights Based Security in the SentryOne Client is a two step process.

1. First, a SentryOne user is associated with either a Windows or SQL Server Authentication account.
2. Next, a SentryOne user or group of users is assigned rights which will restrict server visibility within the SentryOne Client.

You may also associate a Windows Active Directory security group with a SentryOne group. This allows you to easily manage SentryOne Client Security for a large number of users. The Active Directory group can be specified from the group properties tab in the Login field.

1. ASSOCIATE A USER WITH AN ACCOUNT

The first step in configuring Rights Based Security is to associate a SentryOne user with either a Windows account or a SQL Server Authentication account. This account should be the same account that the user uses when opening the SentryOne Client (see also: Connect to Installation). Follow the steps below to associate a user with an account:

1. In the Navigator pane, expand the Contacts node and then the Users node.
2. Double-click the user you wish to assign an account to, or select Open from the context menu. If no user exists you can double-click on the Users node to create a new user.
3. On the properties tab, the Login field is used to specify which account will be assigned rights within the SentryOne Client.
   - If the user connects to the SentryOne Installation using Integrated Windows Authentication, enter their user name in the Login field as Domain\Username.
   - If the user connects to the SentryOne Installation using SQL Server Authentication, simply enter their SQL Server Login name.
4. Save the user, using the Save button found on the toolbar.

2. ASSIGN RIGHTS TO A USER OR GROUP

The Rights tab is located directly beside the Properties tab when editing a user or group. Once a user has been associated with an account, you may then restrict which servers are visible for that user, using the Rights tab. Alternatively, you may choose to assign rights to groups of users. Both users and groups can be assigned a limited set of visible Instances, effectively restricting what the logged-in user can see. Follow the steps below to assign rights to a user or group:

1. From either the User or Group editor, click the Rights tab at the top of the editor. If any restrictions have been previously assigned they will be listed, otherwise the list will be initially empty.
2. At the bottom of the Rights tab, select the Add button to open the list of available Instances.
3. Select the Instances that you wish to configure rights for from the list.
4. The Instances will be added to the Rights tab with checkboxes to Allow or Deny visibility.
5. After saving the changes, the User will only be able to see those Instances in the Rights tab that have Allow visibility (See the note below for more information). If no Instances exist in the Rights tab, the user or group can see all Instances, unless they are a member of another group that has a restricted set of visible Instances.
VISIBILITY OF SERVERS (IMPLICIT VERSUS EXPLICIT DENIAL)

Be aware of the following when assigning rights:

- Once you define rights for a user or group, any Site, Target Group, or Instance which does not have rights explicitly defined, will be implicitly denied to that user or group.
  - i.e. a user or group will be denied visibility for any Site, Target Group, or Instance which is not listed in their rights tab with an Allow checkmark.

When you configure Rights Based Security for Sites and Target Groups also be aware of the following:

- Any Parent node (be it a Site or Target Group) with a Deny permission explicitly configured will override any of its child nodes Allow permissions.
- If a Parent node is being implicitly denied, because rights are not otherwise explicitly defined for it, Allow permissions configured for any of its child nodes will be honored.

ADDITIONAL INFORMATION

Membership of the sysadmin fixed server role is needed to do the following actions (both outlined above):

1. Edit the login associated with a user
2. Assign rights to users and groups

To ensure that a user has rights to login to the SentryOne Database, but does not have rights to modify their own permissions, add the user to the allow_all database role of the SentryOne Database, and ensure the user is not a member of the db_datawriter role. For more information about the available SentryOne Database roles see the Role Based Security topic.

OBJECTS HIDDEN FROM RESTRICTED USERS

These SentryOne Client restrictions are applied to any user with Rights Based Security configured:

The following Navigator Nodes will be hidden/unavailable:

- Contacts *
- Monitoring Service Group
- Monitoring Services
- Object Groups

* Note: The Contacts node will not be hidden if the user is sysadmin for the SentryOne Database

The following commands are unavailable

- Tools menu -> Manage Response Rulesets
- All Targets context menu -> Show System Status
- All Targets context menu -> Show Monitoring Service List

Additionally, Client Alerts and object notes will only be visible for those Instances in which the user has rights.

APPLICATION OF USERS/GROUPS RIGHTS:

Scenario:

- There are 100 servers.
There are 20 users.
There needs to be two different groups Group A (10 Users) and Group B (10 Users) with visibility to different servers.

- **Group A** should have access to servers 1-50
- **Group B** should have access to servers 51-100
- One of the users (AllButOne) should have access to all but one of the servers allowed in Group A and Group B.

**Setup:**

1. Add the 20 users in the Client.
2. Create a **Group A** with rights to servers 1-50 by adding them on the **Rights** tab and checking **Allow**
3. Create a **Group B** with rights to servers 51-100 by adding them on the **Rights** tab and checking **Allow**
4. Add the users to their respective groups on the Groups -> Properties tab.
5. Create user **AllButOne** and add to both groups.
6. On the Rights tab for the **AllButOne** user select **Deny** for the server the user should not be able to access.

*SentryOne* will check groups for restricted Instances first. After the group membership has been evaluated, user Instance restrictions are evaluated. **Note:** The **Deny** overrides any **Allow** that is configured through a group.

### 5.2 Role Based Security

In order to provide a more secure environment and still allow non System Administrators to take advantage of *SentryOne*’s many features, roles are placed on the *SentryOne* Database during its installation or upgrade. Users can be placed in these roles allowing them access to the features they need, while restricting access to other features that may be above and beyond their particular responsibility.

**SETTING UP ROLE BASED SECURITY**

Role Based Security is primarily configured through T-SQL statements or by using SSMS to setup database roles in SQL Server. First make the person a User on the *SentryOne* database and add them to the **allow_all** role. This provides full access to the *SentryOne* database.

From here you can add the User to any of the custom **deny_** roles to restrict that User’s access to the different functions of *SentryOne* . The role names are self-explanatory. Typically there is a role to deny updating the specified information, and one to deny reading the information at all.

**ROLES**
<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_all</td>
<td>Provides full access to SentryOne's features. Place all non-sa Users in this role, then add to deny roles to restrict access.</td>
</tr>
<tr>
<td>deny_actions_read</td>
<td>Denies the ability to view all General, Failsafe, Audit, and Advisory Condition actions.</td>
</tr>
<tr>
<td>deny_actions_update</td>
<td>Denies the ability to make changes to any actions, but allows the viewing of those settings, making them &quot;read-only&quot;.</td>
</tr>
<tr>
<td>deny_appsettings_update</td>
<td>Denies any changes made under the SentryOne Monitoring Service &gt; Settings node.</td>
</tr>
<tr>
<td>deny_contact_update</td>
<td>Denies the ability to update information for individual Users, but allows viewing the information, making it &quot;read-only&quot;.</td>
</tr>
<tr>
<td>deny_contactgroup_update</td>
<td>Denies the ability to update Group information, making it &quot;read-only&quot;.</td>
</tr>
<tr>
<td>deny_customconditions_update</td>
<td>Denies the ability to enable, disable, create, or edit Advisory Conditions.</td>
</tr>
<tr>
<td>deny_eventchain_read</td>
<td>Denies the ability to view Event Chain information.</td>
</tr>
<tr>
<td>deny_eventchain_update</td>
<td>Denies the ability to make changes to Event Chains.</td>
</tr>
<tr>
<td>deny_settings_connection_read</td>
<td>Denies the ability to view information under the Settings tab at the Instance level.</td>
</tr>
<tr>
<td>deny_settings_connection_update</td>
<td>Denies the ability to make changes under the Settings tab at the Instance level.</td>
</tr>
<tr>
<td>deny_settings_object_read</td>
<td>Denies the ability to view information under the Settings tab at the Object level.</td>
</tr>
<tr>
<td>deny_settings_object_update</td>
<td>Denies the ability to make changes under the Settings tab at the Object level.</td>
</tr>
<tr>
<td>deny_settings_source_read</td>
<td>Denies the ability to view Source information from the Settings tab.</td>
</tr>
<tr>
<td>deny_settings_source_update</td>
<td>Denies the ability to make changes to Source information from the Settings tab.</td>
</tr>
<tr>
<td>deny_site_update</td>
<td>Denies changes made to Site Configuration.</td>
</tr>
<tr>
<td>deny_watch_connection</td>
<td>Denies the ability to watch or stop watching an Instance.</td>
</tr>
</tbody>
</table>
deny_watch_object  Denies the ability to watch or stop watching an individual Object.

**AN EXAMPLE**

For example, you may have a junior DBA that needs to be able to use SentryOne's Calendar View to check for any failures or long running jobs overnight, but you don't want them to be able to make changes to any of SentryOne 's settings.

First, add their login as a User on the SentryOne database. Next, place that User in the allow_all role. This will ensure the User has access to all the information they need while being explicitly denied any information specified in the additional roles in which you place them. Finally, for this example, you may want to add this User to all "deny_" roles except the ones ending in ",_read". This will deny changes to any settings along with the ability to watch or stop watching an Instance or object.

It is also important to remember that logins using SQL Server Authentication must be specified in the SentryOne Client Instance information as well. Go to the File Menu -> Connect to Installation, uncheck the box marked "Integrated Windows Authentication", and enter the User's login and password. You will then have to restart the SentryOne Client. These new settings will remain in effect on this SentryOne Client until explicitly changed again.

**Note:** The roles starting with "db." are SQL Server default roles placed on every database. Use of these roles in the SentryOne database may cause unpredictable behavior.
The SentryOne Client provides a seamless experience for server monitoring and optimization, query tuning, alerting, and job scheduling, all within the same thin client interface.

This section discusses the different elements of the SentryOne Client.

- **Client Interface**
  - Navigator Pane
  - Event View Pane
  - Conditions Pane
  - Settings Pane
  - Client Alerts
  - Menus
  - User Preferences
  - License Management

- **Navigator Nodes**
  - Favorite Targets
  - Virtualization
  - All Targets
    - Default Site
    - Targets
    - Instances
  - SQL Server Registrations
  - Contact Management
  - Schedules and Windows
  - Object Groups
  - Monitoring Service
    - Monitoring Service Settings
    - SMTP Configuration
  - Site Configuration
  - Inventory View
  - Monitoring Service Load Balancing and Fault Tolerance
  - Actions Log
  - System Status
  - Service Configuration Utility
  - Monitoring Service Logon Account

- **Common Tasks**
  - Reporting
  - Adding Notes
  - Monitor Additional Instances
  - Connect to Installation
The SentryOne Client interface is divided up into three main areas. The first area, positioned along the left side of the Client by default, contains the Navigator and Event View panes. The second area, positioned along the right side of the Client by default, contains the Actions and Settings panes. The Workspace area occupies the center position. The SentryOne Client has been designed with an enhanced multiple document interface (MDI). All panes are movable, dockable, and resizable. The panes can also be set to auto-hide when not in use to allow for a larger viewable Workspace area.

WORKING WITH PANES

Both the Navigator and Event View panes and the Conditions and Settings panes can be enabled or disabled. In addition, the pushpin feature can be used to auto-hide the panes allowing for a larger Workspace area. These panes can also be docked or undocked allowing the workspace to be customized to best fit your work style. In the event that the panes are closed they can easily be restored from the View menu.

WORKSPACE LAYOUT

The dimensions of the Client Panes and Workspace Area are saved automatically when you close the Client. The default configuration can be restored by selecting the Reset Layout option from the Tools menu.

STATUS BAR

At the bottom of the SentryOne Client is a status bar divided into four sections. Each displays information including:

- The current status for the active view (left)
- The current progress for the active view (middle)
- The current location of the active window relative to the navigation tree (right)
- The Client Alert status indicator (red - errors, yellow - warnings, green - all ok)

Jump to:

- Navigator Pane
6.1.1 Navigator Pane

The Navigator is the tree-view pane positioned on the left side of the client by default. The various nodes within the Navigator are used as the starting point in directing many functions of the SentryOne Client, such as viewing an Event Calendar, or a Performance Analysis Dashboard. The Navigator nodes are also used as the starting point to access the various levels of Conditions and Settings within the Alerting and Response System. Right clicking on a node will access its context menu.

**NAVIGATOR PANE OVERVIEW**

**NAVIGATOR PANE ELEMENTS**

Each object in your SentryOne Enterprise will have an associated icon. Several of the node icons in the Navigator pane will vary based on conditions, such as Instance status, or watched status.

**INSTANCE STATUS**

- A green arrow indicates that the Instance has an **Online** status.
- A red circle indicates that the Instance has an **Offline** status.
- A red exclamation point indicates that the Instance requires attention. Often associated with the SQL Server Agent being offline.
An empty circle indicates that the Instance has a watched status but is not being actively monitored. This can happen if the Monitoring Service has stopped or the Instance belongs to a site without an active Monitoring Service.

**Instance Status** is determined by the SentryOne Monitoring Service’s ability to collect information from the Instance. If an Instance has a green arrow this indicates that the Instance has an Online Status. If an Instance has a red circle this indicates that an Instance has an offline status.

The following is a simplified version of how Instance Status information is obtained and displayed in the SentryOne Client.

1. The SentryOne Monitoring Service first gathers Instance Status information.
2. This Instance Status information is written to the SentryOne database.
3. The SentryOne Client retrieves this Instance Status information from the database and displays it.

You can select any Instance node to access additional status information. A tool tip will display the status (Offline/Online), the Last Monitoring Service Instance Time, and any applicable error information.

**WATCHED STATUS**

A target must first be set to 'Watched' before SentryOne will begin monitoring it. To 'Watch' a target use the right-click context menu of the target. You can also hide unwatched targets so they will not display in the Navigator Pane in the User Preferences (Tools -&gt; User Preferences -&gt; Navigator -&gt; Hide Unwatched Instances). For more information about watching targets and objects see the Watching Instances and Objects topic.

A grayed-out icon indicates that the Instance/object is not being watched.

A colored icon indicates that Instance/object is being watched.

**INSTANCE ICONS**

The different types of Instances are represented with different icons. Refer to the table below.

- SQL Server Instance
- Analysis Services Instance
- Windows Instance

**RED TEXT IN THE NAVIGATOR PANE**

If the label text for a node is red this indicates that an event object has a recent failure. Failures cascade from the object level up to the highest group level containing the object. You can clear a job failure through the right-click context menu of the job, by selecting Cleared Failed Status. You may also adjust how long failures will be displayed in the Navigator through the User Preferences (Tools menu -&gt; User Preferences -&gt; Navigator -&gt; Highlight Event Failures in Navigator for).

**SNOOZING OBJECTS IN THE NAVIGATOR**
All Targets (Global), Sites, Groups, and Targets can be snoozed via a right click context menu in the Navigator pane. Snoozing an object prevents any SentryOne actions from occurring while the snooze is in effect. The snooze status is inherited by objects lower in the hierarchy. For example, snoozing the All Targets (Global) node will effectively stop all SentryOne actions in the environment for the duration of the snooze.

A snooze can be ended early via the right click context menu of a snoozed object.

**NOTE:** Changing the snooze status of an object is logged to the Actions Log via the Audit Condition Snooze Status Changed.

**ADDITIONAL INFORMATION**

For additional information about the specific nodes displayed in the Navigator Pane, see the Navigator topic.

### 6.1.2 Event View Pane

The Event View pane is used to filter what is shown in the Event Calendar view.

When you have a Calendar open to switch to the **Event View**, click the **Event View** tab at the bottom of the pane.

**EVENT VIEW TABS**

The following are descriptions for the various Event View tabs:

- **Filter** - Used for general filter settings including run status, review status, duration, and open text filters.
- **Properties** - Displays a description of the view along with time settings and an object count. For more details see below.
- **Connections** - Displays the Instances and Instance text filters for the current view.
- **Objects** - Displays individual **Event Objects** added to the current view.
- **Event Sources** - Allows you to determine the display properties of each **Event Source**.

**Tab Formatting**

The formatting of the tab text indicates whether certain filters are enabled or disabled. If an event source is deselected, the tab name will be gray. If any objects or categories are selected for a source, the tab name will be in bold. Additionally, any runtime or other filters set on the **Filter** tab will cause that tab name to bold.
Activating Filters

After making filter selections, activate the filter by clicking the **Apply Filter** button near the bottom of the filter menu. When the filter is active, the "Apply Filter" button will have a blue background.

**NOTE:** Runtime filters do not affect status event objects.

Using Text Filters

Multiple keywords are separated with a semicolon (;). A plus sign (+) is used in front of any keyword(s) that you want to apply using an "and" operator. Otherwise the keyword will be applied with an "or" operator. Text filters can be applied to the Event Object Name, the step text, or to the name of the object owner. If filters are entered in two or more of these text boxes, they will be applied with an "and" operator.

Removing Filters

Any filter settings can be removed from the active view by clicking **Remove Filter**. This button is not available on Custom Event Views and other views containing multiple Instances, to ensure a filter is always applied to these views and thus minimize the chance of overloading the SentryOne Client with too much event data.

Saving a Filter as a View

To save the active filter settings to a Custom Event View, select **Save View As** from the File menu.

**PROPERTIES TAB**

The Properties Tab displays basic descriptive information about the current view. Additionally, this tab provides controls to set the default time slice for this view. This allows you to specify the number of days, amount of time, and start time shown for this view every time it is opened. When a Performance Monitor is placed on a Custom Event View, it is applied to all monitorable objects within the view. The View Totals box, at the bottom, allows you to see exactly how many objects will be monitored.

The following are descriptions of the View Totals values.

- **Connections** - The total number of SQL Server instances and Windows Instances associated with the view.
- **Objects From Instances** - Gives a count of all objects from those Instances that meet the filter criteria.
- **Individual Objects** - The number of objects added by the Edit Individual Objects button.
- **Total Objects** - The total objects in the view.
- **Total Monitored Objects** - The total amount of objects that actually have Performance Monitors attached to them.
- **Total Monitorable** - The total number of those objects which can have Performance Monitors attached to them (SQL Agent Jobs, Reporting Services Reports, and Windows Tasks).
This topic covers information related to the **Conditions pane**. For a comprehensive list of all available Actions including information about configuring alerts please see the **Actions** topic which is part of the *SentryOne Alerting and Response System* topic set.

The Conditions Pane is displayed on the right side of the SentryOne Client by default. The Conditions pane is used to configure Actions in response to Conditions being met, as part of the SentryOne Alerting and Response System. Actions can be defined in response to certain **Conditions** being met within your environment. You can choose from a variety of Actions, depending on which Condition is being responded to.

All Conditions work on the principle of inheritance. This means that if you configure an Action in response to a Condition being met at the Global level (**All Targets**), it will be automatically passed down to all applicable objects below it. This allows you to define Global Conditions for the most common issues across your environment once, and have those passed down to every monitored server automatically. You can further refine Conditions at each level as needed. For a visual representation of how inheritance works within SentryOne see the *Alerting and Response System Hierarchy diagram*.

Each Condition that you configure in your environment will have an associated behavior. The behavior controls how the Condition will be carried out relative to any inherited Conditions. There are three Condition behaviors available: Override Inherited Actions, Combine with Inherited Actions, and Disabled. For a complete explanation and example usage scenario for each behavior see the **Action Behavior** section topic.

**VIEW**

The Conditions pane is displayed on the right side of the screen by default. If you do not see the Conditions pane, use the **View** menu (**View -> General Actions/Failsafe Actions/Audit Actions**). Each Condition type has an associated Actions tab where Actions can be configured in response to Conditions being met.

**Conditions pane sections:**

**Inherited** - This section displays the configured Actions that are being inherited by the currently selected object.

- Configured Actions in the Inherited Section have a Status column which indicates if they apply to the currently selected object.
- Configured Actions with a Status of Enabled have Green text.
- Configured Actions with a Status of Disabled are grayed out.

**Explicit** - This section displays the configured Actions that are explicitly set at the current object level.

- An Action that is Explicitly disabled will have Red text.
- Configured Actions with a Status of Enabled have Green text.

**SNOOZING A CONDITION/ACTION**

Alerts can be snoozed or suppressed for a period of time by right clicking on the Condition/Action in the Conditions Pane and selecting one of the following options:
- Snooze this Condition/Action Combination - Snoozes the specific Condition/Action that is selected
- Snooze this Condition for All Actions - Snoozes all instances of the selected Condition regardless of the Action assigned to it
- Snooze this Action for All Conditions - Snoozes all instances of the selected Action regardless of the Condition assigned to it

Additionally, you can select whether the snooze affects just the hierarchical object that you selected (Site, Group, Target, Instance) or it affects all objects.

Conditions that have been snoozed will have a gray background in the Conditions Pane.

Once a Condition/Action is snoozed, it can be unsnoozed by right clicking on the snoozed object and selecting the Unsnooze option. Alternatively, you can unsnooze all snoozed Conditions/Actions through the Tools menu -> Unsnooze All option.

6.1.4 Settings Pane

The Settings pane is where settings are configured for Instances, Sources, and Objects. This tab is also where performance counter thresholds are configured when a counter is selected. Settings can be configured globally, at the Instance level, or at the object level, and just like Conditions they work on the principle of inheritance. See Conditions pane for more information.

**CONNECTION SETTINGS**

Connection settings can be applied at the global level or for an individual Instance. To adjust the settings at the global level, select the devices node and then select the Settings tab to display. There are three types of settings for an Instance; Maintenance Window, Miscellaneous and Synchronization.

**MAINTENANCE WINDOW SETTING**

Maintenance Window settings allow a time frame to be specified where Failsafe Notifications will be disabled while maintenance activities take place on that Instance. This will prevent excessive notifications that may occur if a watched Instance is temporarily inaccessible during these activities. Log Actions will still be enabled during this window, but any Execute or Send Actions will not fire.

**MISCELLANEOUS SETTING (SQL SERVER 2005 AND ABOVE)**

**Auto-enable SQL Server Agent Tokens**

On SQL Server 2005 and above, SQL Server Agent Tokens are disabled as a security measure. They must be enabled to allow SentryOne to watch Alerts on these servers. Selecting True allows SentryOne to automatically enable these tokens when an Alert is watched on a 2005 and above server. More information can be found in SQL Server Books Online.

**Auto-Recycle large SQL Server Agent Logs**

This setting determines whether or not to automatically recycle large agent log files in SQL Server 2005 and above.

**Maximum Queue Length**
This setting specifies the maximum number of jobs that can be queued at one time on this server. See Job Queuing for more information. The default setting is 5.

**Collect Memory by Category Data (SSAS)**

This setting is used to provide more detail than the basic Cleaner memory data, but is higher overhead to collect. Use caution when enabling to ensure it does not impact performance. The default value for this setting is False.

**AZURE SQL DATABASE SETTING**

**Allow SentryOne Monitoring Objects in Target**

In order to provide the best experience, some features of monitoring an Azure SQL Database requires a few small tables and procedures to be stored in the target database. These objects will be created and maintained in a SentryOne schema. If you choose not to allow the monitoring objects to be created, the features dependent upon them (e.g., TOP SQL collection) will be disabled for the target database. This is set to False by default, but can be changed globally or overridden at the target level.

**SYNCHRONIZATION**

**Synchronization Type**

This setting refers to how the SentryOne Client will respond if it discovers the data in the SentryOne Database is not up to date. The default is to show a message box warning of the condition and asking if the SentryOne Client should synchronize the information itself.

**Synchronization Threshold**

The amount of time that must pass since the last successful synchronization by the SentryOne Monitoring Service before the Client attempts to synchronize directly with the target.

⚠️ **IMPORTANT**: This situation may suggest that the SentryOne Monitoring Service is not retrieving data from this Instance. If this is the case, verify that the SentryOne Monitoring Service is running, and check for any entries in the Failsafe log.

**EVENT OBJECT SETTINGS**

To display the Settings for an object type, select the object in either the Navigator or on the Calendar, then select the Settings tab. For each category in the drop-down list for the job, global settings are displayed. This means that it will be automatically inherited by all SQL Agent Jobs unless overridden at the individual Instance or job level.

**NOTE**: Not all event types have object settings, for example, Maintenance Plans or SQL Server Agent Log.

**Queuing Settings**

For SQL Server Agent Jobs only, the Settings screen is used to configure a job’s behavior when queuing another job, or when it is queued by another job. See Job Queuing for more information.

**RUNTIME THRESHOLDS SETTINGS**

In the Runtime Thresholds section, a value for Minimum Runtime Threshold Percent and Maximum Runtime Threshold Percent can be specified. For the global Minimum Threshold, 10%
is usually a good starting value, and for the Maximum Threshold, 250% is usually a good starting value. This means that anytime a job runs for less than 10% or longer than 250% of its average runtime a notification will be sent. If too many messages are being received, these settings can be adjusted as needed.

Explicit time-based thresholds can also be specified. Anytime an explicit time-based threshold is specified it will override the percentage based thresholds for that object. Time-based thresholds are usually less valuable at the global level, particularly the Minimum Runtime Threshold which doesn’t have much value at all globally. Explicit runtime thresholds tend to be more applicable at the actual Instance or object level for overriding the global percentage thresholds on a case-by-case basis.

**NOTE:** One notification will be sent once the Maximum Runtime Threshold has been met and a separate notification will be sent once it has completed.

For example, consider a job that has a great deal of volatility in runtime such as a transaction log backup, which can run for anywhere between 30 seconds and 30 minutes, and its average runtime is 5 minutes. To avoid unnecessary percentage-based threshold notifications for the job, one option would be to set its **Maximum Runtime Threshold** to "35 Minutes" and **Minimum Runtime Threshold** to "20 Seconds". This can be done by selecting either the job’s node in the Navigator or an instance of the job on the calendar, then follow the same steps as above to access and change the job’s runtime threshold settings.

**NOTE:** Runtime Threshold messages for Reporting Services Reports completing in less than two minutes will not be processed until they are complete.

**GENERAL EVENT SOURCE SETTINGS**

**Auto-watch New Objects**

This setting determines whether or not SentryOne will automatically watch any new objects that appear for this event source. **It is highly recommended that Auto-watch is set to True for all sources at the global level**, so that no new objects will slip through the cracks and be missed by SentryOne. The default setting is True.

**Maximum Rows to Synchronize**

Use this setting to adjust the amount of historical data to collect from this source at any given time. This setting is most applicable during the initial synchronization process, to prevent tens or hundreds of thousands of rows from being synchronized from an existing server, which can cause contention issues with the SentryOne database. The default setting is 5000.

**Missed Run Threshold**

The setting determines how much time must pass in order for an object run to be considered missed. The default value for this setting is 5 minutes.

**PERFORMANCE COUNTER SETTINGS**

The **Performance Counter** (‘Monitoring Event Object Performance’ in the on-line documentation) Settings screen is used to set the minimum and maximum values for the counter thresholds of that particular counter. These will be used to trigger the **Performance Counter: Threshold Min** and **Max** conditions if any conditions are enabled for them. This option is only
available when selecting a specific performance counter from the Navigator Pane. There can be only one minimum and one maximum value for a given counter, regardless of how many objects are linked to the counter.

**BLOCKING SQL SETTINGS**

**Collection Settings**

The **Minimum Block Duration** setting determines how long blocking must occur before the blocking condition is triggered. The default for this setting is 15 seconds.

**TOP SQL SETTINGS**

**Runtime Thresholds**

The **Maximum Runtime Threshold** value is used when triggering the Top SQL: Runtime Threshold Max condition. The default value is set to 1 hour at the global level and is set at lower levels to be inherited by the parent. At each level, this value can be overridden or changed.

**TOP SQL SOURCE SETTINGS**

**Collection Settings**

The **Minimum Duration** setting determines the required duration for a Top SQL event to be collected. The default setting is 5 seconds. For safety reasons, it cannot be reduced below 100ms unless one of the CPU, Reads or Writes filters is >0.

The **Minimum CPU, Minimum Reads, and Minimum Writes** settings can also be used to filter the Top SQL events collected. These settings are 0 by default.

The **Collect Trace Events, Collect Statement Events, and Collect Execution Plans** settings specify whether or not these are collected. All are set to True by default.

- If Collect Trace Events is disabled, SentryOne will not start a trace or collect any Top SQL events from a monitored target.
- If Collect Statement Events is disabled, statement-level information will not be available in the Plan Explorer Statements Tree when viewing a query plan.
- If Collect Execution Plans is disabled, Top SQL event collection will occur as normal, however query plan information will not be automatically collected for those events. Plan information can still be collected on demand from within a Plan Explorer session, however.

⚠️ **WARNING**: On systems with very high transaction volumes that also make extensive use of scalar functions, enabling Collect Statement Events may cause a negative impact on the performance of the monitored target, regardless of the effective Top SQL filter settings or the number of events actually collected. See also: Performance impact: SQL trace and user defined scalar functions

The **Running Events Polling Interval** setting specifies how often the service checks for running events, primarily for the purpose of capturing query plan information. This setting does not impact the actual Top SQL events collected. The default setting is 5 seconds.

The **Collect Query Stats** setting enables the collection of query stats. This setting must be set to
true in order for the Query Stats area in the Top SQL tab to work. Note that in order to collect query stats for Azure SQL Database Targets the “Allow SentryOne Monitoring Objects in Target” setting under the Azure SQL Database Connection settings category must also be set to True.

The **Query Stats Sample Interval** specifies how often to sample query stats.

The **Filter Time Span** specifies the base length of time over which the collection filters will be applied to Query Stats.

The **Filter Factor** is calculated by dividing the Query Stats Sample Interval by the Filter Time Span. The collection filters such as Minimum Duration are multiplied by this value when applied to Query Stats collection.

**DEADLOCK SOURCE SETTINGS**

**Collection Settings**

This setting determines if the deadlock statement collection is enabled. The default value is True.

**SQL SERVER AGENT JOB SETTINGS**

**Queuing Settings**

These settings are used to determine the behavior of a job when it is queuing other jobs or when the job is being queued. For more information see [Job Queuing](#).

**SSAS SETTINGS**

**Collection Settings**

The **Minimum Duration** value is the amount of time required for a statement to be logged. The default for this setting is 1 second.

The **Collect MDX/DMX/XMLA Events** setting determines whether or not the command collection is enabled. The default setting is True.

### 6.1.5 Client Alerts

The Client Alert status indicator is located in the bottom right hand corner of the SentryOne Client. When an Alert status is detected, a popup with information about the alert is displayed. You may choose to either pin the alert, dismiss it, or open the Client Alerts window (click on the alert text). Client Alerts are designed to give you actionable information about problems in your monitored environment. To open the Client Alerts dialog, double-click the status indicator in the bottom right of the screen.

You may disable status alert popups or change the recurrence interval through the User Preferences ([Tools menu](#) → **User Preferences** → **System Status**).

The following colors correspond to the different alert levels:

<table>
<thead>
<tr>
<th>Alert Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>A Green status indicates that there are no alerts. (No Alerts)</td>
</tr>
<tr>
<td>Yellow</td>
<td>A Yellow status is the Warning level indicator. Warning level alerts are generated when there is a problem which will limit the full functionality of SentryOne. This could include the SQL Server Agent being offline. (Warning)</td>
</tr>
</tbody>
</table>
Red (Error) A Red status is the Error level indicator. Error level alerts are generated when there is a significant problem in your monitored environment, such as when a Instance has been detected as offline, or if there is a problem synchronizing data.

💡 If you would like to configure notifications for Instance Status and Monitoring Service errors be sure to review the available Failsafe Conditions.

**Concerning the History Synchronization Failed Alert**

If you see the following Client Alert error message concerning history synchronization:

*History Synchronization Failed: Message: SQL Server Agent has information logging turned on. The log will fill quickly in this state. Agent log history synchronization will be paused until information logging is turned off.*

This error indicates that **Information logging is enabled for the SQL Server Agent log**. With this level of logging enabled, the Agent log will quickly grow beyond 1 megabyte in size. The overhead associated with reading the SQL Server Agent log when it grows to this size can cause performance problems. SentryOne will not synchronize the Agent log until **Information logging is disabled**.

Information logging can be turned off as follows:

1. From SSMS (SQL Server Management Studio), right-click the **Error Logs** folder within the SQL Server Agent group node for the associated server, and select the **Configure** command.
2. Uncheck **Information**, and click OK to save.

For more information about SQL Agent log synchronization issues, see this KB article.

### 6.1.6 Menus

The **menus** and **toolbars** are used to operate the SentryOne Client. The options available within the menus will vary depending on the active window. For more information about the toolbars see the Performance Analysis toolbar topic and the Event Calendar toolbar topic.

Following are descriptions of the commands found in the various SentryOne menus.

**The File Menu:**

<table>
<thead>
<tr>
<th>Item</th>
<th>Hotkey</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New Instance</strong></td>
<td>n/a</td>
<td>Add a new Instance to the Navigator Pane.</td>
</tr>
<tr>
<td><strong>New Plan Explorer Session</strong></td>
<td>n/a</td>
<td>Opens a new Plan Explorer Session.</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>n/a</td>
<td>Open Execution Plan files.</td>
</tr>
<tr>
<td><strong>Connect to Installation...</strong></td>
<td>n/a</td>
<td>Specify the SentryOne Database and authentication information for the client.</td>
</tr>
<tr>
<td><strong>Save</strong></td>
<td>Ctrl+S</td>
<td>Save changes to the active window.</td>
</tr>
<tr>
<td><strong>Save View As</strong></td>
<td>n/a</td>
<td>Rename and save the active view.</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Ctrl+W</td>
<td>Close the active window.</td>
</tr>
<tr>
<td><strong>Export Data</strong></td>
<td>Ctrl+E</td>
<td>Export Event List Views and various tabs of Performance Analysis.</td>
</tr>
</tbody>
</table>
### Page Setup

**n/a**

Set the page options for printing.

### Print

**n/a**

Print the active window.

### Print Preview

**n/a**

Preview the active window for printing, exporting and send email options.

### Recent Files

**n/a**

Display a list of recently opened Plan Explorer files.

### Exit

**Ctrl+Q**

Exit the SentryOne Client.

---

#### The View Menu:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigator</td>
<td>View the <a href="#">Navigator Pane</a></td>
</tr>
<tr>
<td>Event View</td>
<td>View the <a href="#">Event View Pane</a></td>
</tr>
<tr>
<td>Plan History</td>
<td>View the <a href="#">Plan History Pane</a></td>
</tr>
<tr>
<td>Conditions</td>
<td>View the <a href="#">Conditions Pane</a></td>
</tr>
<tr>
<td>Settings</td>
<td>View the <a href="#">Settings Pane</a></td>
</tr>
<tr>
<td>Start Page</td>
<td>View the Start Page</td>
</tr>
</tbody>
</table>

#### The Calendar Menu:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Conflicts</td>
<td>Explore conflicts for specified time range.</td>
</tr>
<tr>
<td>View Style</td>
<td>Select the view style for Event Calendar.</td>
</tr>
</tbody>
</table>

#### The Reports Menu:

For more information about the Reporting options in SentryOne see the [Reporting topic](#).

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Calendar</td>
<td>Event Calendar associated reports.</td>
</tr>
<tr>
<td>Performance Analysis</td>
<td>Performance Analysis associated reports.</td>
</tr>
<tr>
<td>General</td>
<td>General reports.</td>
</tr>
<tr>
<td>Report Deployment Settings</td>
<td>SQL Server Report Server location that you wish to deploy your reports.</td>
</tr>
<tr>
<td>Deploy Reports</td>
<td>Deploys the reports to the location specified in the Report Deployment Settings.</td>
</tr>
</tbody>
</table>
Import Reports

Used to import future reports created by the SentryOne development team.

The **Tools** Menu:

<table>
<thead>
<tr>
<th>Item</th>
<th>Hotkey</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Preferences</td>
<td>Ctrl+O</td>
<td>Set the SentryOne Client <a href="#">User Preferences</a>.</td>
</tr>
<tr>
<td>Find Object</td>
<td>Ctrl+F</td>
<td>Object search tool, used to find <a href="#">Event Objects</a> within the SentryOne Client.</td>
</tr>
<tr>
<td>Manage Response Rule sets</td>
<td>n/a</td>
<td>Opens all <a href="#">Response Rule sets</a> for editing.</td>
</tr>
<tr>
<td>Reset Event View Filter</td>
<td>n/a</td>
<td>Restores <a href="#">Event view</a> Filter settings to default.</td>
</tr>
<tr>
<td>Reset Layout</td>
<td>n/a</td>
<td>Restores the Layout to default.</td>
</tr>
<tr>
<td>Download Latest Advisory Conditions Pack</td>
<td>n/a</td>
<td>Downloads latest Advisory Condition pack from SentryOne.</td>
</tr>
</tbody>
</table>

The **Window** Menu:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascade</td>
<td>Cascade active windows in the workspace view.</td>
</tr>
<tr>
<td>Tile Vertical</td>
<td>Tile active windows vertically in the workspace view.</td>
</tr>
<tr>
<td>Tile Horizontal</td>
<td>Tile active windows horizontally in the workspace view.</td>
</tr>
<tr>
<td>Tabbed</td>
<td>Display active windows on tabs in the workspace view.</td>
</tr>
<tr>
<td>Show Tabs on Multiple Line</td>
<td>Allows multiple rows of tabs to be displayed.</td>
</tr>
<tr>
<td>Auto-Rotate Performance Analysis Tabs</td>
<td>Auto-rotates through the active Performance Analysis windows when enabled. The rotation does not change the active tab of a Performance Analysis window. Therefore, whichever tab is selected as active will be refreshed as the windows rotate.</td>
</tr>
<tr>
<td>Active Windows Name</td>
<td>All the Active Windows are listed here.</td>
</tr>
</tbody>
</table>

The **Help** Menu:

<table>
<thead>
<tr>
<th>Item</th>
<th>Hotkey</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SentryOne Help</td>
<td>Ctrl+H or F1</td>
<td>Opens the SentryOne User Guide.</td>
</tr>
<tr>
<td>Item</td>
<td>Hotkey</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SentryOne Quick Start Guide</td>
<td>Ctrl+F</td>
<td>Opens the SentryOne Quick Start Guide.</td>
</tr>
<tr>
<td>Setup Wizard</td>
<td>n/a</td>
<td>Starts the SentryOne Setup Wizard.</td>
</tr>
<tr>
<td>Check for Updates</td>
<td>n/a</td>
<td>Connects with SentryOne's online repository and verifies that you have the latest version.</td>
</tr>
<tr>
<td>Manage Licenses</td>
<td>n/a</td>
<td>Opens the License Entry window where you can view and manage SentryOne Licenses.</td>
</tr>
<tr>
<td>SQLPerformance.com Post</td>
<td>n/a</td>
<td>Create a post on SQLPerformance.com</td>
</tr>
<tr>
<td>Start Remote Support Session</td>
<td>n/a</td>
<td>Enter a Pin and launch a support session.</td>
</tr>
</tbody>
</table>

**Logging**
Options to enable trace logging for the SentryOne Client and Monitoring Service for troubleshooting purposes. Please contact SentryOne support before you enable logging. Logs are stored in the interactive user’s AppData\Local\SentryOne directory.

**About**
Displays current version, licensing information, expiration.

### 6.1.7 User Preferences

To review the user preferences set in your SentryOne Client, select User Preferences from the Tools Menu. The Restore Defaults command will reset all User Preferences to the default value. User Preferences allow you to customize the behavior and appearance of your SentryOne Client.

#### CALENDAR

<table>
<thead>
<tr>
<th>Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calendar Refresh Interval</td>
<td>Set the refresh rate for the Calendar</td>
</tr>
<tr>
<td>Show Events As Conflicts After</td>
<td>Specify the duration that two or more event instances must overlap before being shown as a conflict</td>
</tr>
<tr>
<td>Show Popups For</td>
<td>Set the persistence of popup messages on the calendar</td>
</tr>
<tr>
<td>Default days for Global and Group views</td>
<td>Set the default number of days to display when opening these views</td>
</tr>
<tr>
<td>Persist Calendar Filter Between Views</td>
<td>Select to save the filter settings between views</td>
</tr>
<tr>
<td>Settings</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Auto-scroll Calendar to Current Time</strong></td>
<td>Select to enable Auto-Scroll for the Calendar after there has been no user activity for a period of time.</td>
</tr>
<tr>
<td><strong>Maximum rows to return for calendar/list views</strong></td>
<td>Limiting this value helps to avoid delays in loading views due to a large volume of data.</td>
</tr>
<tr>
<td><strong>Show Running and Historical Events in Recurring Window if Duration Exceeds</strong></td>
<td>This value determines what events will appear in the Recurring Window at the top of the Calendar view.</td>
</tr>
<tr>
<td><strong>Default interval when using Jump To Instance context menus</strong></td>
<td>Set the time interval which will be shown when using Jump To Instance context menus.</td>
</tr>
</tbody>
</table>

### NAVIGATOR

<table>
<thead>
<tr>
<th>Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Highlight Event Failures in Navigator for</strong></td>
<td>Set the length of time to display failed events in red within the Navigator. If you enable the <strong>Until Cleared</strong> option failures will remain in red until they are manually cleared.</td>
</tr>
<tr>
<td><strong>Automatically clear open failures if a subsequent run is successful</strong></td>
<td>If the subsequent run of an object is successful the node will no longer be marked as failed in the Navigator pane.</td>
</tr>
<tr>
<td><strong>Show bold text for failures</strong></td>
<td>Items will become bold and red in the event of a failure.</td>
</tr>
<tr>
<td><strong>Refresh Failures in Navigator Every</strong></td>
<td>Specify how often the Navigator pane will be refreshed to show failures.</td>
</tr>
<tr>
<td><strong>Show Failures for Unwatched Objects</strong></td>
<td>Enable this setting to display unwatched objects in red within the Navigator pane after failure.</td>
</tr>
<tr>
<td><strong>Hide Unwatched Instances</strong></td>
<td>Enable this setting to hide unwatched <strong>Instances</strong> in the Navigator pane.</td>
</tr>
<tr>
<td><strong>Default Event View Style</strong></td>
<td>Select the default view to show when opening an Event Calendar.</td>
</tr>
<tr>
<td><strong>Prompt me to select a product when left-clicking to open a node</strong></td>
<td>Determines whether or not to prompt for the product to use when opening an Instance. When unchecked the default view for device/Instance nodes will be used.</td>
</tr>
<tr>
<td><strong>Default view for Target/Instance nodes</strong></td>
<td>Determines what action SentryOne will take when you click on a Target or Instance node.</td>
</tr>
</tbody>
</table>
**Settings**

**Auto-synch Navigator with the selected tab**
Enable this setting to synch objects selected in the current tab with the Navigator pane.

**Auto-synch Settings/Actions with the selected tab**
Enable this setting to synch objects selected in the current tab with the Settings and Conditions pane.

**Single-click opens new tabs**
Enabling this setting will allow the opening of new tabs with a single click.

**Persist last “Jump To Time” value**
Enable this setting, and each time the Jump to Time context menu item is used, the previous time selection will be highlighted by default.

**Show warning when Alert watch fails due to disabled SQL Agent Alert Tokens.**
If enabled, SentryOne will warn you if you try to watch an Alert on a SQL Server 2005 and above Instance and don't have Agent Tokens enabled.

**Synchronize SQL Server registrations any time a registration group is expanded**
If enabled, SentryOne will synchronize registered servers under the SQL Servers node any time a registration group is expanded. When not checked, the option to “Synchronize SQL Server Groups...” can be used in the Navigator pane to do this.

**Manually manage folders for Reporting Services**
Allows you to create and manage folders beneath the Reporting Services node.

**Default Startup Action**
Determines the first tab displayed on startup.

**COLORS**
Most of the elements in the Event Calendar view may have their colors customized. Beside each color selection is a Pick button. When you select a Pick button, the Color palette is displayed. The Default button is used to reset all colors to their defaults.

**GRAPHS**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum rows to return for graphs on Runtime Stats</td>
<td>Limits the amount of data shown in runtime graphs for performance reasons.</td>
</tr>
<tr>
<td>Maximum rows to return for performance graphs</td>
<td>Limits the amount of data shown in performance graphs for performance reasons.</td>
</tr>
</tbody>
</table>

**UPDATES**
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically check for software updates on startup</td>
<td>When checked, SentryOne will automatically check for updates on startup. Checked by default. An Internet connection is required.</td>
</tr>
<tr>
<td>Automatically check for Advisory Conditions Pack updates</td>
<td>When checked, SentryOne will automatically check for an updated Advisory Conditions Pack when launching the Global level Conditions List. Checked by default. An Internet connection is required.</td>
</tr>
</tbody>
</table>

**SYSTEM STATUS**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Status Refresh Interval</td>
<td>This setting controls how often the system status screen is refreshed.</td>
</tr>
<tr>
<td>Show Successful Tasks for up to</td>
<td>Set the length of time successful tasks will display on the System Status screen.</td>
</tr>
<tr>
<td>Disable Status Popups</td>
<td>If enabled Client alert status message popups will not be displayed.</td>
</tr>
<tr>
<td>Client Alert Recurrence Interval</td>
<td>This setting controls the recurrence interval for alert popups.</td>
</tr>
</tbody>
</table>

**PERFORMANCE ANALYSIS**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show options on Quick Trace launch</td>
<td>When checked, certain options are available when a Quick Trace is launched manually. When unchecked, the last options configured will be used. Options include duration, rows to collect, etc.</td>
</tr>
<tr>
<td>Tab Auto-rotation Frequency</td>
<td>When used in conjunction with &quot;Window &gt; Auto-rotate Performance Analysis Tabs,&quot; it determines how often the tab changes.</td>
</tr>
<tr>
<td>Performance Analysis Overview Refresh Interval</td>
<td>Used to configure how often data is refreshed.</td>
</tr>
<tr>
<td>Default time span for history mode</td>
<td>Defines the History Mode dashboard's default time range.</td>
</tr>
<tr>
<td>Performance Analysis Overview Skin</td>
<td>Used to select the skin color for the Performance Analysis Overview screen.</td>
</tr>
<tr>
<td>Warn on Actual Plan Execution</td>
<td>If this option is enabled, when the Actual Plan command is selected the user will be asked to 'Confirm Query Execution' before the Plan will be collected. You can disable these warnings by unselecting this setting.</td>
</tr>
<tr>
<td>Default Actual Plan Collection Mode</td>
<td>Defines whether or not live profile data is collected by default when an Actual Plan is retrieved.</td>
</tr>
</tbody>
</table>
Live Profile Data Collection Interval

Defines how often live profile data is collected during query execution.

Only save history when command text or connection settings change

If enabled, query plan sessions will only generate history entries when the command text or connection settings are changed.

Allow opening multiple Performance Analysis instances for the same Target using the Navigator

If checked, Performance Analysis tabs will not be reused for the same Target.

Default Link Style

Defines how often live profile data is collected during query execution.

Default Dashboard Mode

Defines the default Dashboard mode when opening Performance Analysis for a target.

Prompt to Save Session Upon Exit

If checked, a prompt will display upon exit to save the active tabs and panes.

6.1.8 License Management

SentryOne is generally licensed per individual monitored target. This includes products for Windows Server, SQL Server, Analysis Services, VMware, and Azure SQL DB. Win Sentry, and V Sentry can be licensed per core. This is most often used in virtualized environments. Hypervisor hosts are licensed based on their number of processor cores, and any virtual machines managed by the host are licensed for all relevant products. Licensing is independent of the SentryOne Client, and there is no limit to the number of Clients which may be installed in your environment. Licensing is also independent of the number of SentryOne Monitoring Services installed in your environment.

ANALYTICS PLATFORM SYSTEM AND SQL DATA WAREHOUSE LICENSING

The licensing for Microsoft Analytics Platform System (APS) and SQL Data Warehouse (SQL DW) differ from the traditional SentryOne licensing model. Monitoring for these types of Targets is dependent on license units. In order to monitor an APS or SQL DW Target, a number of license units equal to or greater than the number of compute nodes on the target is required. If the number of compute nodes exceeds the number of license units, monitoring will be suspended until additional license units are applied. For example, if a six compute node SQL DW target is being monitored with a six node license, then two additional nodes are added to the SQL DW environment to accommodate increased activity, monitoring will stop until an updated license that includes additional license units is applied.

FREE LICENSE OF SQL SENTRY FOR MONITORING THE REPOSITORY
INSTANCE

It is possible to obtain a free license of SQL Sentry for the purpose of monitoring the SQL Server instance that contains the SentryOne database. The free license is available when any of the following is true:

- Five (5) or more SQL Sentry licenses are owned
- One (1) or more (core-based) license is owned

⚠️ IMPORTANT NOTE: The free license will NOT appear in the license count in the Help > About window, whether or not the free license is used. There are no annual software maintenance costs for the free license, and the license is perpetual.

THE HARDWARE KEY

Your SentryOne license has a hardware key which is tied to the location of your SentryOne Database (denoted by the SQL Server instance name). If for any reason you decide to relocate the SentryOne Database, this hardware key can be updated through our Customer Portal by any authorized account holder. As always, you may contact SentryOne support if you have any problems. If you are planning on moving the SentryOne Database, please see the Relocating the SentryOne Database topic.

APPLYING A NEW LICENSE

You can manage your license within the SentryOne Client with the Manage Licenses command, which can be found through the Help Menu (Help menu → Manage Licenses). The Manage Licenses command will bring up the License Entry window.

Select Edit to update the license. Select Clear to clear the existing text. To enter a new license, you can either copy and paste the license text into the window, or drag-and-drop a license file into the space provided. Click Save to apply the license.

The SentryOne Client will be restarted to ensure that the new license settings take effect.

LICENSE OVERVIEW

If you would like to view additional license information such as the number of licenses applied throughout the environment, you can use the About command, which can be found through the Help Menu (Help menu → About).

To get an overview of how your licenses are applied throughout your environment you can also use the Inventory node, which can be found in the Configuration group in the Navigator Pane.

Double-clicking the Inventory node will bring up the targets Workspace Area, which will list every target that SentryOne has ever monitored. A check mark indicates that the target is currently being watched. An empty check box indicates that the target is not being watched.

If you would like to reclaim a license, click on the checkbox to stop watching that target. That license can then be applied to another target as needed.
6.2 Navigator Nodes

The SentryOne Navigator section contains the following topics:

- **Favorite Targets**
- **Virtualization**
- **All Targets**
  - **Default Site**
  - **Targets**
  - **Instances**
- **SQL Server Registrations**
- **Contact Management**
- **Schedules and Windows**
- **Object Groups**
- **Monitoring Service**
  - **Monitoring Service Settings**
    - **SMTP Configuration**
  - **Site Configuration**
  - **Inventory View**
  - **Monitoring Service Load Balancing and Fault Tolerance**
  - **Actions Log**
  - **System Status**
  - **Service Configuration Utility**
  - **Monitoring Service Logon Account**

6.2.1 Favorite Targets

Each SentryOne Client has its own unique **Favorite Targets group**. The Favorite Targets group is used to define a view of Targets and Instances specific to that individual SentryOne Client. This is useful in a large environment, as it allows an individual user to easily specify a subset of Targets and Instances which they are responsible for. You may toggle displaying the Favorite Targets node with the **Show Favorite Targets** button located directly above the **Navigator pane**.

When you add a **Target** or an **Instance** to your Favorite Targets, if it is not already a part of your SentryOne Environment, it will be added to the **Default Site** of the **All Targets** node. When you add a Target or Instance to your Favorite Targets that is already a part of your shared SentryOne Environment, the Target or Instance will not be duplicated. As mentioned earlier, the Favorite Targets group is unique to each SentryOne Client and define a view of your larger SentryOne Environment.
Changes you make regarding the **Watched status**, or to any configured Settings, Conditions, and Actions from the context of the Favorite Targets group node will be mirrored in the **All Targets** node. In the same way, changes made to Targets and Instances from the context of the All Targets node will also be mirrored back to any Favorite Target groups which they belong.

### 6.2.2 Virtualization

The Virtualization node houses the support for VMware and Hyper-V virtualization options.

**VMWARE**

To add a vCenter, right-click on the VMware node and select the *Add vCenter* option. Enter the server name and display name, then either enter your credentials or check the box to use SSPI. You can also select which Site will be responsible for the monitoring of the vCenter.

In order to add a vCenter, the Monitoring Service must be a configured user on the vCenter implementation and must have read-only access at the minimum.

**NOTE:** If using SSPI authentication, be sure to select a Site where the Monitoring Service has at least read-only access to the vCenter.

**HYPER-V**

Whenever a Hyper-V Guest or Host is monitored with SentryOne, it will appear under the Hyper-V node. If the Host is monitored with SentryOne, all of that Host's Guests will appear as well. If only a Guest is watched with SentryOne, that Guest's Host will be displayed.

### 6.2.3 All Targets

The **All Targets** node is the root node of the SentryOne Client and can be found in the top position of the **Navigator Pane**. All Targets that are registered across the entire SentryOne enterprise can be found within **All Targets**.

The **All Targets** node can be used to configure Global Conditions and Settings within the SentryOne **Alerting and Response System**. The All Targets node can also be used to view the Event Global View Calendar or a Performance Analysis Shared Targets overview.

**All Targets Context Menu**

Right click on the Shared Groups node to access its context menu. The following commands are available:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open → Event Calendar</td>
<td>Opens the Event Manager Global View of your SentryOne enterprise.</td>
<td><a href="#">Event Calendar</a></td>
</tr>
<tr>
<td>Open → Performance Overview</td>
<td>Opens the Performance Analysis Shared Targets Overview.</td>
<td><a href="#">Performance Analysis Overview</a></td>
</tr>
<tr>
<td>Add Site</td>
<td>Adds a new Site node beneath the Shared Groups Node.</td>
<td><a href="#">Site Configuration</a></td>
</tr>
</tbody>
</table>
GLOBAL LEVEL OF THE ALERTING AND RESPONSE SYSTEM

The All Targets node is used to configure Global Conditions and Settings within the SentryOne Alerting and Response System. When the All Targets node is selected in the Navigator pane, the associated global level Conditions and Settings pane will be available. If you do not see the Conditions and Settings pane when the All Targets node is selected, use the View Menu to bring up the desired pane. Conditions and Settings configured at the All Targets level are global in nature. This means that when a Condition or Setting is set at the Shared Group level it will automatically propagate down throughout your SentryOne environment.

Example:

You configure a Send Email Action for the SQL Agent Job: Failure condition at the All Targets global level.

An email will now be sent anytime an Agent Job fails across your entire monitored environment.

Once Global Conditions and Settings are configured, more granular configurations can be made where needed at the lower levels. For more details on how Global Conditions and Settings are inherited throughout your SentryOne environment, and information about setting up alerting for your SentryOne installation, see the Alerting and Response System topic.

GLOBAL VIEWS

You can access Global Views of your SentryOne enterprise through the All Targets node. This can be done by either double clicking the All Targets node or by using its right-click context menu. The default number of days for the Global Calendar view is one day, and can be adjusted in the User Preferences.

The Primary purpose of the Global Views is give you insight into what is happening across your enterprise without having to open individual Event Calendars or Performance Analysis dashboards at the target level. Future and running events are disabled in the Event Calendar Global view, but you can adjust the runtime filter and other filter criteria on the Event View filter tab to further refine the display for personal preferences. For more information about the Performance Analysis Shared Targets Overview, see the Performance Analysis Overview topic.

ALL TARGETS CHILD NODES

Jump to:

→ Default Site
→ Targets
→ Instances
6.2.3.1 Default Site

Sites represent a logical grouping of Targets, Instances, and Monitoring Services within your SentryOne environment. If you are upgrading from a previous version of SentryOne you will notice that the Sites displayed beneath the All Targets node directly reflect those sites you previously configured within the Site Configuration node. If you are a new user you will notice that the Instances you added using the Quickstart wizard are all contained in the Default Site.

When a Site node is selected in the Navigator pane the associated Site level Conditions and Settings pane will be available. If you do not see the Conditions and Settings pane when the Site node is selected, use the View Menu to bring up the desired pane. Conditions and Settings configured at the Site level apply to all Targets, Instances, and Objects in that Site. This means that when a Condition or Setting is configured at the Site level it will automatically propagate down throughout that Site.

Site Context Menu

Right click on any Site node to access its context menu. The following commands are available:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Target</td>
<td>Adds a new Target to the selected Site.</td>
<td>Targets</td>
</tr>
<tr>
<td>Add Target Group</td>
<td>Adds a new Target Group to the selected Site.</td>
<td>Targets</td>
</tr>
<tr>
<td>Add Connection</td>
<td>Adds a new Connection to the selected Site.</td>
<td>Connections</td>
</tr>
<tr>
<td>Delete Site</td>
<td>Deletes a Site from the Shared Group node. Any Targets or groups contained within the Site must first be deleted before a site can be deleted. This command is not available from the Default Site node.</td>
<td>n/a</td>
</tr>
<tr>
<td>Rename Site</td>
<td>Renames the currently selected Site.</td>
<td>n/a</td>
</tr>
<tr>
<td>Open Performance Analysis</td>
<td>Opens a Performance Analysis Overview for every Target in the Site.</td>
<td>Performance Analysis Overview</td>
</tr>
</tbody>
</table>

**MONITORING SERVICES**

Beneath every Site node there is a Monitoring Services node. This Monitoring Services node contains the Monitoring Service(s), identified by their associated Target name(s), that are responsible for collecting information pertaining to that site.

ℹ️ The SentryOne Monitoring Service is a Windows Service which is responsible for collecting event history, event status, and performance related information and storing that information
The Monitoring Service(s) within a Site are responsible for gathering information about those watched Targets and Instances contained within that same site. Available Monitoring Services can be assigned to Sites either through the Site Configuration node or by dragging and dropping a Monitoring Service within the Navigator pane. This enables you to control exactly which Monitoring Services watch which Targets, and can be critical for scenarios where you have multiple servers located at different geographic locations. Having a Monitoring Service for each geographic location minimizes network traffic across the WAN.

For more information see the Monitoring Services and Site Configuration topics.

**TARGETS**

Beneath every Site node, are its associated Target nodes, identified by the Target's name. Each Target node represents a physical computer, commonly a Windows Server, within your SentryOne Environment. Contained beneath every Target node are those associated Instances that reside on that Target.

Within the framework of the Alerting and Response System Hierarchy, Targets represent a level between Sites and Instances. This means that when a Condition or Setting is configured at a Target level it will automatically propagate down to its associated Instances, and objects.

If you do not see the Conditions and Settings pane when a Target node is selected, use the View Menu to bring up the desired pane. Conditions and Settings configured at the Target level apply to all Instances, and Objects beneath that Target. This means that when an Condition or Setting is configured at the Target level it will automatically propagate down to its associated Instances, and objects.

For more information see the Targets topic.

**INSTANCES**

Contained within every Target node are its associated Instance nodes. Instances are identified by their Target/Instance name. Contained beneath every Instance node are its associated objects.

Within the framework of the Alerting and Response System Hierarchy, Instances represent a level above objects. This means that when a Condition or Setting is configured at an Instance level it will automatically propagate down to its associated objects.

If you do not see the Conditions and Settings pane when an Instance node is selected, use the View Menu to bring up the desired pane. Conditions and Settings configured at the Instance level apply to all objects belonging to that Instance. This means that when a Condition or Setting is configured at the Instance level it will automatically propagate down to its associated objects.

For more information see the Instances topic.
6.2.3.2 Targets

Contained within every Site node are its associated Target nodes. Target nodes are identified with the Target's full name, usually ComputerName.DomainName. Each Target node represents a physical computer, commonly a Windows Server within your SentryOne Environment. Beneath every Target node in the Navigator pane are its associated Instances.

Double click any Target node to access its related Performance Analysis Dashboard or Event Calendar.

**Targets Context Menu**

Right click on any Targets node to access its context menu. The following commands are available:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open → Event Calendar</td>
<td>Opens the Event Manager Calendar View for the associated Target.</td>
<td>Event Calendar</td>
</tr>
<tr>
<td>Open → Performance Analysis</td>
<td>Opens the Performance Analysis dashboard for the associated Target.</td>
<td>Performance Analysis Dashboard</td>
</tr>
<tr>
<td>Show Windows Processes</td>
<td>Launches a Process Activity Window similar to Windows Task Manager for the associated Target.</td>
<td>Windows Process Activity</td>
</tr>
<tr>
<td>Add Instance</td>
<td>Adds a new Instance to the Default Site.</td>
<td></td>
</tr>
<tr>
<td>Delete Target</td>
<td><strong>Warning:</strong> This command will Delete the actual registration for the Target from SentryOne. This includes the deletion of all historical information for any associated connections belonging to the Target from the SentryOne Database. This command is final and cannot be undone. Only use this command if you want to permanently delete information SentryOne has stored about the Target and its associated Connections from the SentryOne Database.</td>
<td></td>
</tr>
<tr>
<td>Jump to Time</td>
<td>Allows you to select a date, time of day, and time frame for an Event Manager Calendar</td>
<td></td>
</tr>
<tr>
<td>Scan for Configuration Changes</td>
<td>This scan looks for any hardware, network, or schema changes that have taken place on the Target or its associated Instances. Note, this information is automatically gathered periodically.</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**TARGET CONDITIONS AND SETTINGS**

Within the framework of the Alerting and Response System Hierarchy, Targets represent a level between Sites and Instances. This means that when a Condition or Setting is configured at a Target level it will automatically propagate down to its associated Instances and objects.

If you do not see the Conditions and Settings pane when a Target node is selected, use the View Menu to bring up the desired pane. Conditions and Settings configured at the Target level apply to all Instances and objects beneath that Target. This means that when an Condition or Setting is configured at the Target level it will automatically propagate down to its associated Instances and objects.
ADDING A TARGET TO YOUR ENVIRONMENT
To add a new Target to your installation you can use the Add Target command found in the right-click context menu of any Site Node.

DELETING A TARGET FROM YOUR ENVIRONMENT
To delete a Target from your installation use the Delete Target command found in the right-click context menu of any Target node. Please see the following warning concerning the Delete Target Command.

Groups
Targets can be organized into Target Groups beneath any Site node. A Target Group allows you to create logical groups of devices for alerting, organizational, and display purposes. The Add Target Group command can be found in the right-click context menu of any Site node.

TARGET GROUP CONDITIONS AND SETTINGS
Within the framework of the Alerting and Response System Hierarchy, Target Groups represent an intermediate level between the Site and Target. This means that when a Condition or Setting is configured at a Target Group level it will automatically propagate down to its associated Targets, Instances, and objects.

6.2.3.3 Instances
Contained beneath every Target node is its associated Instance nodes. Instances are identified by their Target/Instance name. Contained beneath every Instance node are its associated objects.

INSTANCES ACTIONS AND SETTINGS
Within the framework of the Alerting and Response System Hierarchy, Instances represent the level above objects. This means that when a Condition or Setting is configured at an Instance level it will automatically propagate down to its associated objects.

When an Instance node is selected in the Navigator pane the associated Instance level Conditions and Settings pane will be available. If you do not see the Conditions and Settings pane when an Instance node is selected, use the View Menu to bring up the desired pane. Conditions and Settings configured at the Instance level apply to all Objects belonging to that Instance. This means that when a Condition or Setting is configured at the Instance level it will automatically propagate down to its associated objects.

ADDING AN INSTANCE TO YOUR ENVIRONMENT
You can easily add additional monitored Instances to your SentryOne environment. This is accomplished by right-clicking either the All Targets node, a Site node, a Target Group node, or an existing Target Node in the Navigator pane and using the Add Instance command. You can also
add an Instance through the File menu.

In the Add Instance dialog you may choose the desired Instance Type from the drop-down menu (Analysis Services Instance, SQL Server Instance, Windows Instance). See also: Monitor Additional Instances

### 6.2.4 SQL Server Registrations

The SQL Server Registrations Node mirrors the SQL Server registrations of the native client tools for the current user including any SSAS registrations, and Central Management Servers. Any groups or servers that are added or deleted through the client tools will be automatically added to the SQL Server Registrations Node in the SentryOne Client.

You may toggle displaying the local SQL Server Registrations with the Show SQL Server Registrations button located directly above the Navigator pane. Changes you make regarding the Watched status, or to any configured Settings, Conditions, and Actions from the context of the SQL Server Registrations node will be mirrored in the All Targets node.

Activities normally done in SQL Server's client tools, such as adding, editing, or deleting jobs, alerts, or maintenance plans can be done using the SQL Server Registrations node in the SentryOne Client. Servers can also be registered from the SentryOne Client through the right-click context menu of the Local Server Groups node.

As mentioned, SentryOne also includes support for accessing registered servers managed by a Central Management Server. Just like other Instances in the SQL Server Registrations group, changes you make regarding the Watched status, or to any configured Settings, Conditions, and Actions from the context of the Central Management Servers group will be mirrored in the All Targets node.

### 6.2.5 Contact Management

The Contacts Navigator node is used to create and maintain Users and Groups within your SentryOne environment. Users and Groups are created for couple of different reasons. Firstly, they can be created for notification purposes within the SentryOne Alerting and Response System. When you configure a Send Email or Send Page action, any user or group which has been created will be available for you to choose as a target.

The second reason for defining Users and Groups is the ability to assign rights to them. These rights are used to restrict the access of users and groups within the SentryOne Client. For more information, please see the SentryOne Client Security topic.

**VIEWING NOTIFICATIONS BY USER OR GROUP**

From the Contacts node it is also possible to view all notifications assigned to a particular user or group. To do so select any user or group node and open the Conditions pane (View Menu → General/Failsafe/Audit Actions). A list of all notification actions associated with that user/group will appear; from here you can disable or change any of these configured Actions.

**USERS**
Expand the **Users** node in the Navigator pane to list any configured users.

**ADD OR EDIT A USER**

To add a new user simply double-click on the Users node. To edit information about an existing user, double click on the user’s name in the Navigator to open the *Edit User* tab.

**DELETE OR DISABLE A USER**

You may also delete or disable a user through the right-click context menu. When you disable a user, they will no longer receive notifications for any actions which they are the selected target of. If the disabled user is part of a group, that user will also no longer receive notifications when that group is notified. Disabled users are grayed-out in the Navigator and also have *(disabled)* next to their name. To enable a user that has been disabled, use the right-click context menu of the user and select the Enable User command.

When you create a new user or edit an existing user the following fields will be available.

**USER FIELDS**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>The first name of the user. <strong>Required.</strong></td>
</tr>
<tr>
<td>Last Name</td>
<td>The last name of the user.</td>
</tr>
<tr>
<td>Email Address</td>
<td>The email address of the user. <strong>Required</strong></td>
</tr>
<tr>
<td>Pager Address</td>
<td>The pager address of the user.</td>
</tr>
<tr>
<td>Email Window</td>
<td>The email window for the selected user. When you assign an Email Window to a user or group be sure to take note of the Email Window Behavior option as detailed below. For more information about schedules please see the Schedules and Windows topic.</td>
</tr>
</tbody>
</table>

**Perform actions that occur within the window** - Only email Actions which occur during the Window's active time frame will be carried out. email Actions which are triggered outside of the active time frame are suppressed.

**Don't Perform actions that occur within the window** - Email Actions which occur during the active time frame are filtered or suppressed. Only email Actions which occur outside of the Window's active time frame are carried out.

The pager window for the selected user. When you assign a Pager Window to a user or group be sure to take note of the Pager Window Behavior option as detailed below.

For more information about schedules please see the Schedules and Windows topic.

**Perform actions that occur within the window** - Only pager Actions which occur during the Window's active time frame will be carried out.
Pager Window Behavior

Don’t Perform actions that occur within the window - Pager Actions which occur during the active time frame are filtered or suppressed. Only pager Actions which occur outside of the Window’s active time frame are carried out.

The Windows or SQL Authentication account for the user. You can associate a SentryOne user with a Windows or SQL Authentication account and restrict access to specific Sites, Target Groups, or Instances effectively limiting what the logged-in user can see within the Client. For more information, please see the SentryOne Client Security topic.

Description

Descriptive text or notes for the user.

Groups

The Available area lists any group within your SentryOne enterprise.
The Selected area lists those groups which the user belongs to.

GROUPS

Expand the Groups node in the Navigator pane to list any configured groups.

ADD OR EDIT A GROUP

To add a new group simply double-click on the Groups node. To edit information about an existing group, double click on the group name in the Navigator to open the Edit Group tab.

ASSIGN USER(S) TO A GROUP

To assign users to the group, choose from the available users on the bottom-left side of the view, and click the Add button. Click on the Save button in the top toolbar to save a group.

DELETE OR DISABLE A GROUP

You may also delete or disable a group through any group’s right-click context menu. When you disable a group, that group will no longer receive notifications for any actions which they are the selected target of. Disabled groups are grayed-out in the Navigator and are also listed as disabled. To enable a group that has been disabled use the right-click context menu of the group and select the Enable Group command.

GROUP FIELDS

When you create a new group or edit an existing group the following fields will be available.

Field | Description
--- | ---
Group Name | The name to assign to a group.
Alert Window | The alert window for the selected group. When you assign an Alert Window to a group be sure to take note of the Alert Window Behavior option as detailed below.
For more information about schedules please see the Schedules and Windows topic.

**Perform actions that occur within the window** - Only email and pager Actions which occur during the Window's active time frame will be carried out. Actions which are triggered outside of the active time frame are suppressed.

**Don’t Perform action that occur within the window** - Actions which occur during the active time frame are filtered or suppressed. Only email and pager Actions which occur outside of the Window’s active time frame are carried out.

The Windows Active Directory account for the group. You can associate a SentryOne group with a Windows Active Directory account and restrict access to specific Sites, Target Groups, or Instances effectively limiting what the logged-in user can see within the Client. For more information, please see the SentryOne Client Security topic.

**Description**

Descriptive text or notes for the user.

**Users**

The Available area lists users within your SentryOne enterprise. The Selected area list any users that belong to the group.

6.2.6  Schedules and Windows

SentryOne employs extensive scheduling capabilities through its Schedules and Windows features. Schedules and Windows can be managed through their respective nodes found in the Navigator pane.

**Schedules** have the following applications within SentryOne:

- Fragmentation Manager Operations

**Windows** have the following applications within SentryOne:

- Maintenance Windows
- Email and Pager Windows for Users and Groups
- Ruleset Windows for Configured Actions

**SCHEDULES**

SentryOne allows you to create Schedules that can be applied in the following places:
Scenario

When you enable **Fragmentation Manager** in your SentryOne environment you will select a schedule to perform Fragmentation Manager operations. This schedule is initially specified in the Fragmentation Manager wizard. After you enable Fragmentation Manager for an Instance you may access the applied schedule through the **Index Defragmentation Settings** found in the Settings Pane (View -> Settings).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
</table>

- **Index Defragmentation Settings**, including the applied Schedule, can be configured at the following levels: All Targets (Global), Site, Target Group, Target, Instance, database, table, and individual index.

See Also: **Fragmentation Manager**

**WINDOWS**

SentryOne allows you to create Windows that can be applied in the following places:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
</table>

- **Maintenance Windows** allow a time frame to be specified while daily maintenance activities take place. When a **Failsafe Condition** occurs during a Maintenance Window only the associated log actions will be completed, any configured send email actions will be suppressed.

There are several options for configuring maintenance windows within SentryOne, including maintenance windows specific to Targets, Instances, and the SentryOne Database. For more information please see the [Eliminating Failsafe alerts using Maintenance Windows](#) KB article.

- **Email and Pager Windows** can be assigned to a user or group. Windows control the time frame of when alerts are sent to the user or group. When you assign an **Email or Pager Window** to a user or group you have the option to change the **Window Behavior** to one of the following:

  **Window Behavior**

  - **Perform actions that occur within the window** - Only email and pager Actions which **occur during** the Window's active time frame will be carried out. Email and pager Actions which are triggered outside of the active time frame are suppressed.
**Email and Pager Windows for Users and Groups**

**Scenario**

**Don’t Perform actions that occur within the window** - Email and pager Actions which occur during the active time frame are filtered or suppressed. Only email and pager Actions which occur outside of the Window's active time frame are carried out.

💡 To apply an Email or Pager Window to a user or group, first double-click the user or group in the Navigator pane. This will open an *Edit User/Group tab*. Use the *Email Window* or *Pager Window* drop-down list box to choose the desired Window. Use the *Email Window Behavior* or *Pager Window Behavior* drop-down list to choose the desired behavior.

See also: [Contact Management](#).

**Windows** may be applied to configured Actions which will control the time frame of when that Action can take place. When you assign a **Window to a configured Action** you have the option to change the **Window Behavior** to one of the following:

**Window Behavior**

- **Perform actions that occur within the window** - Only email and pager Actions which occur during the Window’s active time frame will be carried out. Email and pager Actions which are triggered outside of the active time frame are suppressed.

- **Don’t Perform actions that occur within the window** - Email and pager Actions which occur during the active time frame are filtered or suppressed. Only email and pager Actions which occur outside of the Window's active time frame are carried out.

💡 To apply a Window to a configured action, first select the action in the *Conditions pane*. Next open the *Ruleset tab* for the selected action. Use the *Window* drop-down list to select your desired Window. You may also choose to edit an existing window with the *Edit* button, or create a new window with the *New* button. Use the *Window Behavior* or *Window Behavior* drop-down list to choose the desired behavior.

See Also: [Alerting and Response System](#).

**Ruleset Windows for configured Actions**

**CREATING A SCHEDULE OR WINDOW**

To create a new Schedule or Window first expand the Schedules and Windows node in the Navigator pane. Next, double-click either the Schedules or the Windows node to open the
respective dialog screen. You may edit or delete an existing Schedule or Window by selecting it, and then choosing the **Edit** or **Delete** command. To create a new Schedule or Window select the **New** command.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This is the name which will be applied to the schedule or window. <strong>Recurring</strong> - A recurring schedule or window will be repeated according to the frequency and duration you choose. <strong>One Time</strong> - A one time schedule or window will take place only on the occurrence date you specify. <strong>Compound</strong> - A compound schedule or window can be created which combines previously created schedules.</td>
</tr>
<tr>
<td>Schedule Type</td>
<td>Specify the duration or the length of time that the Window be active. <strong>Note:</strong> Window Duration applies only to Windows and is N/A for Schedules. Select the time you would like the one time occurrence of the schedule or window to take place. <strong>Note:</strong> This option is only applicable for Schedule Type: One Time.</td>
</tr>
<tr>
<td>Frequency</td>
<td>The Frequency of the schedule or window can be set to either Daily, Weekly, or Monthly. Additional frequency options will be available depending on the occurrence type you choose. <strong>Daily Frequency</strong> Use the Daily Frequency options to select the specific time(s) during the day the schedule or window will occur. <strong>Duration</strong> Use the Duration section to specify a Start Date and End Date. <strong>Summary</strong> The Summary description will auto-fill according to the options you choose. You may add to the summary, or change the text as desired.</td>
</tr>
</tbody>
</table>

### 6.2.7 Object Groups

**Object Groups** are useful when you need to quickly apply like policies to a set of objects that do not exist within the same hierarchical group in your SentryOne environment. These objects could be dispersed throughout your environment, in different sites, groups, or subtrees, but if they share a similar set of characteristics, they may be ideal candidates for a similar set of policies.

Take for example, all of the transaction log backup jobs within your environment. You may wish to suppress runtime threshold notifications generated when these jobs run, by disabling the send
email Action. **Object Groups** allow you to do this without affecting any other Agent Jobs. In the past, this would involve you having to go and touch each of the transaction log backup job's settings individually. **Object Groups** allow you to avoid the tedious process, saving you time, as you can quickly locate objects by name or category, add them to a group, and apply a set of policies.

As **Object Groups** exist outside of the normal SentryOne Hierarchy, a **Condition** or **Setting** configured for an Object Group will be applied last, after any inherited or explicitly defined Condition or Setting. In this way, **Object Group** Conditions and Settings have the final say within your environment.

As mentioned earlier, typical uses include disabling runtime threshold alerts for all transaction log backup jobs, and applying policies to QA servers that may exist within several sites of your environment. When you create an **Object Group** you will choose how configured Conditions and Settings will be applied in the group. You can create an **Object Group** in which configured Conditions and Settings apply strictly to the group members. Alternatively, you may create an **Object Group** in which configured Conditions and Settings apply to the group members and their children. See the Creating a New Group section below for more information.

Each **Object Group** you create also has an associated Evaluation Order. This Evaluation Order is specific to just the set of Object Groups in your environment. This is valuable when you have an object which belongs to multiple Object Groups with a divergent set of policies. The Evaluation Order allows you to control which group’s configured Conditions and Settings will be applied. A configured Setting or Condition from the **Object Group** with the highest Evaluation Order will be applied last, and therefore will be the effective Setting or Condition.

When you add new objects to an **Object Group**, any Setting which is applicable to that Object type will be configurable in the Settings pane. By default, each object in an **Object Group** will still receive its base set of inherited Settings from its parent objects in the SentryOne Hierarchy. If you would like to override these Settings, you must first define a Setting within the **Object Group** itself. To do this you will need to change the Defined Setting to true. After you define a Setting for an **Object Group**, that Setting will be reflected in any applicable objects which belong to that group.

Just like Settings, Condition/Action combinations become available as appropriate when objects are added to a group. With any **Object Group** selected you can configure Conditions through the Conditions pane, just as you normally would. See the Configuring Conditions and Settings for an **Object Group** section below for more information.

### Setting up Object Groups

To create a new **Object Group** first double-click the **Object Groups** node in the Navigator pane. This will open the **Object Groups Editor** window. The Groups section of the editor window contains any previously configured groups in your environment. The Group Members section of the editor window allows you to view individual members of any selected **Object Group**.

### Creating a New Group

To create a new **Object Group** use the Add command in the Groups section of the editor window. This will open the Add Group dialog window. Choose a name and a description for the group. Also take note of the Applies To drop-down box. See below for reference

**Applies To:**
**Selected group members and their children**

Any Conditions or Settings you configure for the group will also apply to the children of the group members. This makes the group's Settings and Conditions inheritable.

**Selected group members only**

Any Conditions or Setting you configure for the group will only apply to the specific members of the Group and will have no effect on their children. This makes the group's Settings and Conditions noninheritable.

If you are creating an **Object Group** to set policies for all of the QA servers in your environment, you will want to choose the *Selected group members and their children* option, as this would give you access to applicable Settings and Conditions from the Instance level down to the object level.

If you are creating an **Object Group** to set policies for just a finite set of objects, and do not wish to impact or have access to children of the object, use the *Selected group members only* option. Given the transaction log backup job use case discussed earlier, it would not actually matter which *Applies To* option you chose, as the Agent job object is a leaf node in the SentryOne Hierarchy.

**ADDING OBJECT GROUP MEMBERS**

To add new members to a group, first select a group in the **Groups section** of the editor window. Next, in the **Group Members section** of the editor window use the **Add** command.

In the **Select Group Members** dialogue screen, first choose the object type of the group member you would like to add from the tree view. Next, use the **Search** command to view objects of that type in your monitored environment. Select an object and use the **OK** command to add it to the Group. You may add multiple objects by using **Ctrl** or **Shift** select.

**CONFIGURING CONDITIONS AND SETTINGS FOR AN OBJECT GROUP**

Once you have added your desired objects to the group, you can begin configuring the Conditions and Settings for the group. You configure Conditions and Settings just like you normally would, through the Conditions and Settings pane. If you do not see the Conditions and Settings pane use the **View** menu to open it.

To add an Condition to a Group, first select the **Object Group**. This can be done in either the **Navigator pane** or through the **Object Groups Editor** screen. From the **Conditions pane**, use the **Add** command to open the Actions Selector window. Choose your desired Condition/Action combinations and then select the **OK** command. Any Conditions that are applicable to the objects within your group will be available for you to configure.

You may also configure Settings for the **Object Group** through the **Settings pane**. Any Settings that are applicable to the objects in your group will be available for you to configure. As discussed earlier, by default each object in an **Object Group** will still receive its base set of inherited Settings from its superior parent objects in the SentryOne Hierarchy. If you would like to override these Settings, you must first define a Setting within the **Object Group**. To do this you will need to change the *Defined* flag to true in the Settings pane. After you define a Setting for an **Object Group**, that Setting will be reflected in any applicable objects which belong to that group.
If you are ever unsure of where an object is receiving its inherited Settings from you can easily check. The Active Settings List Report will list this information for each Settings in the Object Name column. Access this report from the Reports Menu (Reports -> General -> Active Settings List).

Alternatively, you can view this information directly at the individual object level. First select the desired object in the Navigator and open the Settings pane. The Inherent From Parent section of each setting group will contain the name of the superior object which is passing down that group of Settings. In the case where the object's Setting have been defined with an Object Group it will list that Object Group as the parent.

6.2.8 Monitoring Service

The SentryOne Monitoring Service, a Windows service, collects event history, status, and performance information, sends notifications, and performs various other tasks related to event management for each “watched” Instance. The various event metadata and history information collected by the SentryOne Monitoring Service is stored in a SQL Server database. The number of Instances that can be watched in your environment is controlled by the number of licenses you have purchased.

SENTRYONE CONFIGURATION NODE

- Monitoring Service Settings
  - SMTP Configuration
- Site Configuration
- Inventory View
- Monitoring Service Load Balancing and Fault Tolerance
- Actions Log
- System Status
- Service Configuration Utility
- Monitoring Service Logon Account

6.2.8.1 Monitoring Service Settings

The SentryOne Monitoring Service Settings are global in nature, applying to the entire SentryOne installation. They control the behavior and function of all installed SentryOne Monitoring Services connected to the active SentryOne database. The SentryOne Monitoring Service is a Windows service installed on one or more servers in your enterprise. All settings are stored in the SentryOne database. The SentryOne Monitoring Service Settings can be accessed through the Settings node found in the Monitoring Service group of the Navigator pane. By default the Monitoring Service
Settings will open to the SMTP Configuration tab.

**MONITORING SERVICE SETTINGS AND DATA RETENTION**

There are also Monitoring Service Settings which control the retention of certain types of data within the SentryOne Database.

- The retention of *Event Calendar* data (Event History data) is controlled with the *Event History Monitor* "Purge History Older Than" setting. The default value for this setting is 30 days.

- The retention of data displayed on the *Performance Analysis Top SQL, Blocking SQL, and Deadlocks* tabs (Performance Monitor data) is controlled with the *Performance Monitor* "Purge History Older Than" setting. This setting also applies to Event Calendar *General Performance Monitoring (on-line documentation)* data. The default value for this setting is 15 days.

  > **Note:** Top SQL, Blocking SQL, and Deadlock data will also be converted to the native Event Calendar format and then retained according to the Event History Monitor "Purge History Older Than" setting.

- All other Performance Data, including the data which is displayed on the *Performance Analysis Dashboard*, is not affected by the Monitoring Service Settings. Instead, this data is subject to a high performance storage scheme and internal rollup structure as outlined in the [Dashboard Resolutions/Retention topic](#) and the [Data Capacity Planning article](#).

**SMTP CONFIGURATION**

For information about configuring an SMTP server please see the [SMTP Configuration](#) topic.

**EVENT MONITOR**

This tab is used to configure settings for the process responsible for polling watched *Instances* for active event status information.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polling Interval</td>
<td>Controls how often the SentryOne Monitoring Service will poll each watched Instance for event status information. The polling operations used are lightweight, however, the value can be increased to reduce network activity associated with the SentryOne Monitoring Service. The default Polling Interval is 15 Seconds.</td>
</tr>
<tr>
<td>Error Resume Interval</td>
<td>Controls how long the SentryOne Monitoring Service should wait to recover from an error or exception in the event monitor, when attempting to obtain event status information from an Instance. The default Error Resume Interval is 1 Minute.</td>
</tr>
<tr>
<td>Trigger &quot;Run Missed&quot; Condition</td>
<td>Use the check-box to specify whether or not to execute the Run Missed Condition when the Instance is offline.</td>
</tr>
<tr>
<td>Minimum Block Duration Required to Trigger &quot;SQL Server Agent Job: Block Condition&quot;</td>
<td>Used to specify the minimum time a SQL Agent Job must be part of a blocking chain before triggering the SQL Server Agent Job: Block Condition.</td>
</tr>
</tbody>
</table>

**EVENT HISTORY MONITOR**
This tab is used to configure settings for the process responsible for the collection of event history information from watched Instances.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polling Interval</td>
<td>Controls how often the SentryOne Monitoring Service will poll each watched Instance for event history information.</td>
</tr>
<tr>
<td>Error Resume Interval</td>
<td>The Error Resume Interval is how long the SentryOne Monitoring Service should wait to recover from an error or exception in the event monitor when attempting to obtain event status information from an Instance. Controls the retention of Event Calendar data (Event History data). The default value for this setting is 30 days.</td>
</tr>
<tr>
<td>Purge History Older Than</td>
<td>Note: Retention of Event Calendar General Performance Monitoring (on-line documentation) data is controlled with the Performance Monitor “Purge History Older Than” setting.</td>
</tr>
<tr>
<td>Max Thread Pool Size</td>
<td>This is an advanced setting that may improve performance in large (100+ watched Instances) environments. This setting tells chaining how far back in the history to look when determining whether history entries should affect chain status. This setting is used as a fail safe measure if for some reason you were unable to synchronize with a server for a specified period of time, whenever the server comes back online and SentryOne is able to re-synchronize history information, any history that is older than the time specified by this setting will not be used to impact chain execution status. The minimum value for this setting is 10 minutes.</td>
</tr>
<tr>
<td>Event Chains are Only Affected by History Within</td>
<td>Note: It is recommended that you contact SentryOne support before changing the default value.</td>
</tr>
</tbody>
</table>

MESSAGE PROCESSOR

These settings control the process that polls the message queue and processes messages. Examples of processes include logging to disk or the event log, or sending an email notification.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polling Interval</td>
<td>Polling Interval controls how often the SentryOne Server’s internal message queue is checked and messages are processed. Increasing this setting will effectively lengthen the time between when a condition is met and any associated actions are triggered. Adjusting the setting up or down will affect overhead on the SentryOne Monitoring Service only, and will not affect network or monitoring overhead for watched Instances.</td>
</tr>
<tr>
<td>Error Resume Interval</td>
<td>The Error Resume Interval is how long the SentryOne Monitoring Service should wait to recover from an error or exception in the message processor.</td>
</tr>
<tr>
<td>Do Not Fire Actions for Messages Older Than</td>
<td>If messages were unable to be collected for some time this setting will keep actions from being fired for these older messages.</td>
</tr>
<tr>
<td>Purge Action Log History Older Than</td>
<td>This setting determines how many days worth of Action Log information is stored.</td>
</tr>
</tbody>
</table>

These settings determine whether global message throttling is enabled and, if so, how many email messages can be sent within the provided amount of time before the message throttling activates. When message throttling activates the "Message Processor: Message Action Throttled" Condition is fired.
**Global Message Throttling**

(Do’t send more than X emails within X hours.

The "Message Processor: Message Action Throttled" condition is a failsafe condition; therefore by default any user added during the Setup Wizard will be notified when this Condition is met. If you disable alerting for the "Message Processor: Message Action Throttled" Condition, Message Throttling will still remain in effect.

**Note:** The "Message Processor: Message Action Throttled" Condition is subject to global message throttling itself.

---

### PERFORMANCE MONITOR

These settings control the process responsible for monitoring Windows performance counters that are linked to event objects. The "Purge History Older Than" setting also controls retention of Performance Analysis Top SQL, Blocking SQL, and Deadlock data.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Polling Interval</strong></td>
<td>Polling Interval controls how often the SentryOne Monitoring Service will sample any performance counters linked to an event. The polling is not continuous, but is started and stopped automatically based upon an event object’s schedules.</td>
</tr>
<tr>
<td><strong>Error Resume Interval</strong></td>
<td>The Error Resume Interval is how long the SentryOne Monitoring Service should wait to recover from an error or exception in the performance monitor.</td>
</tr>
<tr>
<td><strong>Purge History Older Than</strong></td>
<td>Specifies how long Performance Analysis Top SQL, Blocking SQL, and Deadlock data is retained in the SentryOne Database. This setting also applies to Event Calendar General Performance Monitoring (on-line documentation) data.</td>
</tr>
</tbody>
</table>

**Note:** All other Performance Data, including the data which is displayed on the Performance Analysis Dashboard, is not affected by the Monitoring Service Settings. Instead, this data is subject to a high performance storage scheme and internal rollup structure as outlined in the Dashboard Resolutions/Retention topic and the Data Capacity Planning article.

**Note:** Top SQL Blocking SQL and Deadlock data will also be converted to the native Event Calendar format and then retained according to the Event History Monitor “Purge History Older Than” setting.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Text Data Maximum Length</strong></td>
<td>Change the maximum length of textdata captured for Top SQL and Top Command events.</td>
</tr>
<tr>
<td><strong>Text Data (Normalized) Maximum Length</strong></td>
<td>Change the maximum length of normalized text data captured for Top SQL and Top Command events.</td>
</tr>
<tr>
<td><strong>Normalization Engine</strong></td>
<td>With the release of version 8.2, there have been improvements made to the Normalization Engine. Users upgrading from a prior build will need to select the Advanced option to see the enhancements.</td>
</tr>
</tbody>
</table>

---

### MAINTENANCE WINDOW

...
Maintenance window settings allow you to specify a time frame where Failsafe Notifications will be disabled while daily maintenance activity takes place. This will prevent excessive notifications that may occur if a watched Instance is temporarily inaccessible during these activities. Log Actions will still be enabled during this window, but any Execute or Send Actions will not fire inside the Maintenance Window. The Maintenance Window specified in this view, under the SentryOne Monitoring Service node, applies to activity on the SentryOne Database itself. To specify a Maintenance Window for watched Instances / Target, select the Settings tab from the Settings Pane for that Instance / Target.

SNMP CONFIGURATION

SNMP Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>This value determines where the SNMP trap will be sent. The default setting (BROADCAST) will send the trap to all managers. Individual managers can be sent the traps using the individual hostname or IP address.</td>
</tr>
</tbody>
</table>

Note: Managers included in the BROADCAST group and who are also listed individually will receive two messages for each occurrence.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>The port used for transmission of the traps (Default 162).</td>
</tr>
<tr>
<td>SNMP Version</td>
<td>The version of SNMP to be used (1, 2C or 3)</td>
</tr>
<tr>
<td>Enabled</td>
<td>This checkbox determines whether or not the trap is enabled.</td>
</tr>
</tbody>
</table>

SNMP v1/2C Configuration

Community
Enter the community.

SNMP v3 Configuration

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine ID</td>
<td>Enter the Engine ID.</td>
</tr>
<tr>
<td>Username</td>
<td>Enter the username</td>
</tr>
<tr>
<td>Privacy Algorithm</td>
<td>Select the appropriate privacy algorithm</td>
</tr>
<tr>
<td>Privacy Password</td>
<td>Enter the privacy password</td>
</tr>
<tr>
<td>Hash Algorithm</td>
<td>Select the appropriate hash algorithm (MD5, SHA, or None).</td>
</tr>
<tr>
<td>Hash Password</td>
<td>Enter the password.</td>
</tr>
</tbody>
</table>

6.2.8.1.1 SMTP Configuration

The SMTP Server configuration options can be accessed through the Monitoring Service Settings (Navigator pane - Monitoring Service node - Settings - SMTP Config tab).
Use the drop-down list box to choose a previously configured SMTP server, or use the (...) command to open the Select an SMTP Server dialog, and configure a new SMTP server.

The selected SMTP server will be the globally defined SMTP server, and will be used for all notification purposes unless it is overridden at the Site level. For more information about this, please see the Configuring a Site Level SMTP Server section below.

**ADD A NEW SMTP SERVER**

From the SMTP Config tab, use the (...) command to open the Select an SMTP Server dialog. To add a new SMTP server, first use the New command to open the SMTP Server Editor dialog box. See the table below for more information about the available options. Once you have entered the required information use the Test command to send a test email.

---

**Note:**

- The SMTP Security Settings are optional, and are not required in most environments.
- For the most accurate SMTP test, you should use a SentryOne Client installed on the SentryOne Monitoring Service machine to send a test message. If you use a different computer, i.e., your local workstation, the results may be different. For example, your SMTP server may allow relay from your workstation's subnet, but not from the SentryOne Server's subnet. In this case the test from your workstation would succeed; however, the SentryOne Monitoring Service would be unable to deliver notifications.

---

### SMTP Configuration Options

<table>
<thead>
<tr>
<th>SMTP Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTP Server</td>
<td>Enter the IP Address or the Domain Name of the SMTP Server. When using localhost, this will be the local SMTP server on the machine where the SentryOne Monitoring Service is installed.</td>
</tr>
</tbody>
</table>

| From Address | Specify the From Address for Email notifications |

<table>
<thead>
<tr>
<th>SMTP Security (Optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Enable SSL</td>
</tr>
<tr>
<td>Use Custom Port</td>
</tr>
</tbody>
</table>

**CONFIGURING A SITE LEVEL SMTP SERVER**

SentryOne allows you to configure multiple SMTP servers for the different Sites within your environment. This allows you to specify an SMTP server, or an SMTP Server group which is local for each Site, simplifying permissions, and reducing traffic over WANs.

You may configure Site level SMTP servers through the Site Settings found in the Settings pane. To do this first select your desired Site in the Navigator pane. Next open the Settings pane (View menu -&gt; Settings). In the Settings pane choose Site Settings from the top drop-down list. Change the Inherit From Parent Setting to false. The SMTP Server General setting will list the
current SMTP server. Next, click on the current SMTP server and select the (...) command to open the Select an SMTP Server dialog. Choose your desired SMTP server and select the OK command to specify the SMTP server for your Site.

**SMTP SERVER GROUPS**

In order to provide failover capabilities, SentryOne provides you with the ability to create SMTP Server groups. To create an SMTP Server group first add your SMTP servers to your SentryOne environment, as outlined above in the add a new SMTP Server section.

Next you will need to create the SMTP Server group. From the Select an SMTP Server dialog box, use the New command to open the SMTP Server Editor dialog box. Use the Type drop-down list and choose Server Group. Give the Server group a name and add your desired servers with the Add command.

**SMTP SERVER GROUP FAILOVER PROCESS**

The SMTP servers in a group will failover in sequence, according to the order in which they are listed in the SMTP Server Editor dialog box. You may change this order by using the arrow buttons located at the bottom of the dialog box.

**An SMTP server fails**

When an SMTP server fails, the Monitoring Service will attempt to use the next SMTP server in sequence to send the message. If the Monitoring Service is successful in using this SMTP server, this same SMTP Server will be used the next time the Monitoring Service needs to send a message.

For example, assume you have a Send Email Action configured for the SQL Server Agent Job: Failure Condition, and also you have the following SMTP servers defined sequentially in your group:

1. SMTP_A
2. SMTP_B
3. SMTP_C
4. SMTP_D

When the failure Condition is detected the Monitoring Service will attempt to complete the Send Email Action using SMTP_A, which is the first server in the group. If the Monitoring Service is unable to send the message through SMTP_A, it will try SMTP_B. If it is successful, the Monitoring Service will use SMTP_B the next time it needs to complete a Send Email Action.

**Every SMTP server fails**

In the event that every SMTP server in the group is unavailable when the Monitoring Service attempts to complete a Send Email Action, a Monitoring Service: Action Failed Condition will be triggered. Given our example SMTP group above, this process would work as follows:

If the Monitoring Service is unable to send the email message through SMTP_A, it will try SMTP_B, then SMTP_C, and finally SMTP_D. If the Monitoring Service completes a cycle through each SMTP server in the SMTP Server group, and is still unable to complete the Send Email Action, a Monitoring Service Action: Failed Condition will be triggered.

**Resetting the Failover process**

Changes made to an SMTP Server group, including changes to the evaluation order, or additions and deletions of SMTP servers will effectively reset the failover process. Once the failover process
has been reset, the Monitoring Service will attempt to process Send Email Actions starting with the first SMTP server defined for the group.

6.2.8.2 Site Configuration

Sites are logical groupings of Targets, Instances, SMTP Servers, and Monitoring Services within your SentryOne environment. Every SentryOne environment has at least one Site configured by default; you can see this Site represented in the Navigator pane, as the Default Site. While not every environment will have a need for multiple Sites, they offer a number of advantages for environments which contain geographically sparse assets, existing in multiple domains. See the SentryOne Enterprise Architecture (‘Enterprise Architecture ’ in the on-line documentation) topic for a graphical representation of this idea. The following section contains general information about Sites, if you prefer you can jump right to the walkthrough section.

SITE LEVEL MONITORING SERVICES

The SentryOne Monitoring Service is a Windows Service, which is responsible for collecting event history, event status, and performance related information about your servers, and storing that information in the SentryOne Database.

SentryOne Monitoring Services can be assigned to specific Sites within your environment. Once you place a Monitoring Service within a Site it becomes exclusively responsible for gathering information about those watched Targets and Instances contained within that same Site.

As you can imagine, having a Monitoring Service within the same LAN as those servers it is monitoring significantly reduces WAN traffic, as the Monitoring Service is able to consume data locally and write back to the SentryOne Database as needed.

SITE LEVEL SMTP SERVERS

As discussed above, once a Monitoring Service is assigned to a Site that Monitoring Service becomes responsible for the Site in a number of ways. One of the responsibilities of the Monitoring Service is to alert you about events which are happening within your environment. One of the ways it does this is by communicating with your SMTP servers, when sending email alerts. If you have a Site in New York and another in Chicago, it doesn’t make sense to use New York’s SMTP server to alert you about Conditions happening in Chicago.

Site level SMTP Servers and SMTP Server Groups solve this problem. When the Monitoring Service detects a Condition which you have configured to be notified about, it will package that alert and then marshal that information to the local SMTP server, which you have assigned to the Site. This all happens within the locality of the Site’s LAN, again avoiding long distance communication over a WAN, and significantly reducing network traffic.

Site level SMTP servers can be configured through the Site Settings. For more information about this, see the SMTP Configuration topic.

SITE LEVEL POLICIES

SentryOne allows you to set policies for your monitored environment by configuring Settings and Conditions/Actions. Each of these Settings and Conditions has a hierarchical configuration. The basic idea is that you set global policies and then reconfigure them as needed at the different levels available within the SentryOne Hierarchy. As you may have guessed, Sites represent a level within the SentryOne Hierarchy. This offers a number of possibilities, as you can fine tune your Settings and Conditions/Actions with just the Site level environment in mind.
CONFIGURE A NEW SITE WALKTHROUGH

Take the example of a company with offices in both Atlanta and Louisville, with five monitored servers at each location. Using separate Sites for Atlanta and Louisville, a SentryOne Monitoring Service can be installed in Louisville which monitors only the five servers there, and which reports back to the SentryOne Database in Atlanta. This configuration will result in less network traffic across the WAN link than if the SentryOne Monitoring Service in Atlanta watched the Louisville servers. Additionally, both the Atlanta and Louisville Sites could be configured to use their own local SMTP server or SMTP server group, further reducing WAN traffic.

CONFIGURE A SITE THROUGH THE NAVIGATOR PANE

SentryOne v7 added the ability to configure Sites through the Navigator pane, simplifying the process. This example walks through creating a site through the Navigator pane. The same general configuration options are also available through the Site Configuration Window. Find that example directly below this one.

1. Right-click on the All Targets node in the Navigator pane and use the Add Site command.
2. The new Site will be created beneath the default Site within the Navigator Pane; give the Site a name.
3. Install the new SentryOne Monitoring Service on a non-production server in the remote Site, following the guidelines below.

   **Important:** In this step, when you are installing the Monitoring Service in your remote Site, be sure that the installation package build number is identical to the version of your existing installation. Your SentryOne version can be viewed in the Help → About menu of the SentryOne Client.

   a. When prompted for the SentryOne Database information, specify your existing SentryOne Database and choose your desired Authentication Mode. If there is no trust between the domains, you will need to use either SQL Server authentication or pass-through authentication to write back to the SentryOne Database.

   b. When prompted for the Service Account Information be sure to use an account with the required permissions, namely SysAdmin for each monitored SQL Server, and Windows Administrator privileges for any servers you plan to watch with SentryOne.

4. Wait just a brief amount of time for the Monitoring Service to completely initialize. The Monitoring Service will initially be displayed in the Monitoring Service Group of the Default Site. It will be identified with the name of the Target it was installed on.
5. Select the newly installed Monitoring Service and drag it to the Monitoring Service folder of the Remote Site.
6. Add any Instances and Targets that you wish to monitor through the right-click context menu of your newly created Site.

CONFIGURE A SITE THROUGH THE SITE CONFIGURATION WINDOW

1. Open the Site Configuration window (Navigator pane → Monitoring Service → Site Configuration).
2. In the Sites Section use the **New button** to open the *Add/Edit Site dialog*.

3. Give the Site a name and select **OK**.

4. Install the new SentryOne Monitoring Service on a non-production server in the remote Site, following the guidelines below.

   **Important:** In this step, when you are installing the *Monitoring Service* in your remote Site, be sure that the installation package build number is identical to the version of your existing installation. Your SentryOne version can be viewed in the *Help → About* menu of the SentryOne Client.

   - When prompted for the **SentryOne Database information**, specify your *existing SentryOne Database* and choose your desired Authentication Mode. If there is no trust between the domains, you will need to use either *SQL Server authentication* or *pass-through authentication* to write back to the SentryOne Database.
   - When prompted for the **Service Account Information**, be sure to use an account with the *required permissions* for monitoring servers in the Remote Site, namely SysAdmin for each monitored SQL Server, and Windows Administrator privileges for any servers you plan to watch with SentryOne.

5. Wait just a brief amount of time for the *Monitoring Service* to completely initialize, then click off and back on the Site Configuration node to refresh the window.

6. Select the new Site in the Sites area of the Site Configuration window. The new SentryOne Monitoring Service should show up as Available. Select it and click **Add**, then click **Save**.

7. The newly created Site and its associated Monitoring Service will be displayed in the Navigator pane. Add any Instances and Targets that you wish to monitor through the right-click context menu of your newly created Site.

   **Note:** The Default Site acts as a "catchall" for any SentryOne Monitoring Services and Targets that are removed from or haven’t yet been added to any other Sites.

### 6.2.8.3 Inventory View

The Inventory view allows you to see the **Watched Status** of any Target within your environment. You can drag the column header to the top to group by that column. Other options are available in the context menu for the column headers for grouping, sorting, filtering etc. Once applied, the filter will be shown at the bottom of the pane.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>Management Site maintained through the Site Configuration screen, or through Navigator pane drag and drop functionality. Sites represent a logical grouping of Targets, Instances, and Monitoring Services within your SentryOne environment.</td>
</tr>
<tr>
<td>Monitoring Service</td>
<td>Specific SentryOne Monitoring Service by machine name.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of specific Target.</td>
</tr>
<tr>
<td>Type</td>
<td><strong>Targets</strong> may be SQL Server, Analysis Services, Windows, APS Appliance, VMware Hosts, Azure SQL DB, or SQL DW</td>
</tr>
<tr>
<td>Watched</td>
<td>The flags indicate whether the Target is currently being watched by</td>
</tr>
</tbody>
</table>
SentryOne. A check mark indicates that the Target is currently being watched. An empty check box indicates that the Target is not being watched. If you would like to reclaim a license, click on the checkbox to stop watching that Target. That license can then be applied to another Target as needed.

Use the Inventory view to quickly see how the licenses within your environment are distributed.

6.2.8.4 Monitoring Service Load Balancing and Fault Tolerance

In order to provide automatic load balancing and fault tolerance SentryOne allows for the installation of multiple Monitoring Services.

There is no additional configuration required to implement this functionality. Once an additional SentryOne Monitoring Service is installed and connected to the same SentryOne database during setup, they will automatically distribute the monitoring load evenly between themselves providing automatic fault tolerance and load balancing. If one SentryOne Monitoring Service fails, the remaining SentryOne Monitoring Services will pick up the load automatically.

For example, if there are three SentryOne Monitoring Services associated with nine servers all contained within the same Site, the SentryOne Monitoring Services will divide the servers evenly among themselves, each managing three. If one SentryOne Monitoring Service were to fail, the other two would pick up the unmanaged servers and divide the remaining three between themselves.

6.2.8.5 Actions Log

SentryOne includes an interface for viewing, sorting, and filtering log and notes information. The Actions log is where information relating to General, Failsafe, and Audit Actions can be viewed. Additionally whenever a Condition is configured with the Log to Database Action, information related to that Condition can also be viewed in the Actions Log.

LOG FILTER

Filter Criteria used to manage results displayed in the view. Use the Log Filter to search for a specific log entry by narrowing the selection criteria to be displayed in the Log List.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>After</td>
<td>Enables the filter to only show logs with an event time after the specified time in the date/time selection controls.</td>
</tr>
<tr>
<td>Before</td>
<td>Enables the filter to only show logs with an event time before the specified time in the date/time selection controls.</td>
</tr>
<tr>
<td>Object Name</td>
<td>Filters the log list by applying a like comparison to the specified value to the value stored in the Object Name field.</td>
</tr>
<tr>
<td>Parent Name</td>
<td>Filters the log list by applying a like comparison to the specified value to the value stored in the Parent Name field.</td>
</tr>
<tr>
<td>Contains</td>
<td>Narrows the log list to the entries with the specified value in the message text.</td>
</tr>
<tr>
<td>Log Type</td>
<td>Specify the type of Actions to display in the Log List.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Filters for a specific condition category.</td>
</tr>
<tr>
<td>Action</td>
<td>Filters for a specific action category.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Used to refresh the list of logs and date/time filter.</td>
</tr>
<tr>
<td>Apply Filter</td>
<td>Used to apply Filter Criteria settings and refresh the log list.</td>
</tr>
<tr>
<td>Reset Filter</td>
<td>Used to remove the filter applied to the log list.</td>
</tr>
<tr>
<td>Page</td>
<td>Used to show the current display page of results and how many pages are in the log list.</td>
</tr>
<tr>
<td>First</td>
<td>Move to the first page of log results</td>
</tr>
<tr>
<td>Previous</td>
<td>Move to the previous page of log results.</td>
</tr>
<tr>
<td>Next</td>
<td>Move to the next page of log results.</td>
</tr>
<tr>
<td>Last</td>
<td>Move to the last page of log results.</td>
</tr>
</tbody>
</table>

**LOG LIST**

The Log List shows all log records retrieved. Records cannot be edited, however, notes can be added. To sort a list, click on the header.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Time</td>
<td>Displays the time the action was fired for the event.</td>
</tr>
<tr>
<td>Event Time</td>
<td>Displays the start time for the event.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Displays the object type for the event.</td>
</tr>
<tr>
<td>Object Name</td>
<td>Contains the name of the object for the event.</td>
</tr>
<tr>
<td>Parent Object Name</td>
<td>Contains the name of the parent object for the event.</td>
</tr>
</tbody>
</table>
Condition
Captures the condition that caused the event.

Action
Captures the action that caused the event.

NOTES LOG LIST
Even though notes are considered a logged object, the properties are somewhat different than the logs for other events. Thus, the log list columns vary in order to reflect these differences.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Time</td>
<td>Date/time note was created.</td>
</tr>
<tr>
<td>User Name</td>
<td>Name of user that created the note.</td>
</tr>
<tr>
<td>Object Name</td>
<td>The object name that the note references.</td>
</tr>
<tr>
<td>Parent Object Name</td>
<td>The name of the parent object that the note references.</td>
</tr>
<tr>
<td>Note Text</td>
<td>Contains the note message text.</td>
</tr>
<tr>
<td>Note Title</td>
<td>Contains the title or subject of the note.</td>
</tr>
<tr>
<td>Calendar</td>
<td>Captures the condition that caused the event.</td>
</tr>
<tr>
<td>Notifications</td>
<td>Captures the action that caused the event.</td>
</tr>
</tbody>
</table>

MESSAGE WINDOW
The Message window is located at the bottom portion of the view, and contains a read-only area, displaying the log messages. To add or edit notes to an event instance, right-click on the desired row in the list, and choose the option to **Add/Edit Notes** to open the note editor.

6.2.8.6 System Status

The **System Status** screen provides information about the work SentryOne is doing in the environment, including Monitoring Service tasks, their current state, and any associated messages. By default the **System Status** screen will only display errors. Generally, there is no need to reference the **System Status** screen, but it may be helpful when troubleshooting environmental faults.

You may choose to view the information displayed in **System Status** grouped by either Target or Monitoring Service with their associated commands.

Be aware that turning off show errors will trigger the SentryOne database to elevate logging. This can temporarily increase the transactions on the database until the System Status view is closed.

6.2.8.7 Service Configuration Utility
The Service Configuration Utility is used to change the stored credentials of the SentryOne Monitoring Service. Changing the security account using the services applet in the Windows control panel will cause the service to fail and is not supported. The service will appear to be running, but will not be able to perform any work due to an encryption error.

**Running the Service Configuration Utility is the only supported way of changing the stored Monitoring Service credentials.** If only the password needs to be changed, the services applet in the Windows control panel can be used.

The "ServiceConfiguration.exe" utility is located in the SentryOne program group on the start menu.

6.2.8.8 Monitoring Service Logon Account

If the User account used by the Monitoring Service needs to be changed, the SentryOne Service Configuration Utility will need to be run in order for the Public/Private key encryption to validate the change.

For more information about the different requirements for the Monitoring Service logon account please see the Monitoring Service Security topic in the Quick Start Guide.

6.3 Common Tasks

6.3.1 Reporting

SentryOne offers an expansive list of reporting options. SentryOne reports are generated using SQL Server Reporting Services (SSRS) technology. The SentryOne Client includes Reporting Services Report Viewer which allows the SentryOne Reports to be viewed from within the SentryOne Client. All reports can be accessed from the Reports menu.

SentryOne Reports can also be deployed to a Reporting Services 2005 and above server for distribution.

The Report Deployment Settings are accessed from the Reports menu. For more information about these settings and deploying SentryOne Reports please see the Report Deployment section of this topic.

**REPORT PARAMETERS**

Once a report is selected, depending on the type of report, the user can specify a number of parameters including a date and time range for the report. Options may also be available to save these parameters for later use. To save a set of parameters, choose the Save command on the Report Parameters dialog screen. To load a saved set of parameters simply choose a parameter set from the Load Saved Parameters drop-down list. For certain reports, such as the SQL Server Performance report, an option will be available to select multiple sets of parameter values. Use the Add to Parameter List command to add a parameter set to the selected report.

**EXPORTING REPORTS**
SentryOne Reports can be exported in a variety of formats for distribution purposes. After a report is run, from the Report Viewer screen, choose the Export button found on the toolbar as shown below to export a report:

![Export button](image)

### EVENT MANAGER REPORTS

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Failures</td>
<td>The <strong>Current Failures</strong> report provides data about all failures which have occurred in the given time frame for any monitored objects within your SentryOne Environment. The report includes various information about the associated failures including: name of the event, parent Instance name, last step output, start and end times. Report parameters allow you to choose to show either Open Failures only or All Failures. Any user notes added to the Job failure instance will also be available on this report. Use the <strong>Current Failures</strong> report to quickly identify object failures over the selected time frame.</td>
</tr>
<tr>
<td>Management Summary</td>
<td>The <strong>Management Summary</strong> report provides a high-level overview of the monitored servers within your SentryOne environment. The report includes a count of all monitored objects, and as applicable, the associated number of failures recorded for those objects, grouped in the following ranges: last 30 days, last 7 days, and last 24 hours. Use the <strong>Management Summary</strong> report to help you determine which servers are the most active in your environment.</td>
</tr>
<tr>
<td>Performance Counters List</td>
<td>The <strong>Performance Counters List</strong> report provides data about performance counters (‘Schedule Performance Monitoring‘ in the on-line documentation) currently associated with specific event objects or custom event views within your SentryOne Environment. Information about each performance counter is displayed for the specified time range, including the minimum, maximum, and average of the recorded metric. Use the <strong>Performance Counters List</strong> report to quickly identify active performance counters in your environment.</td>
</tr>
</tbody>
</table>
**PERFORMANCE ANALYSIS REPORTS**

The Performance Analysis reports are the companion to SentryOne Performance Analysis. For more information about the various metrics contained in the Performance Analysis reports please review the Performance Metrics topic.

Note: On applicable reports, the Load Saved Parameters drop-down box allows the user to save the parameters for future use.

**PERFORMANCE**

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global Server Performance</strong></td>
<td>The Global Server Performance report provides a summary of the performance metrics collected for each of the servers in your SentryOne environment. Use the Global Server Performance report in conjunction with the Performance Analysis Global View for a high-level understanding of server performance across your entire monitored environment.</td>
</tr>
<tr>
<td><strong>Index Defragmentation Details - By Database</strong></td>
<td>The Index Defragmentation Details report provides a summary of the defragmentation details for the indexes in your environment. Defragmentation metrics provided include fragmentation percentages, defrag method, fragmentation percentage post defrag, and start/end times.</td>
</tr>
<tr>
<td><strong>Index Fragmentation and Usage</strong></td>
<td>The Index Fragmentation and Usage report provides a graphical representation of the overall fragmentation, disk space usage and buffer usage of the indexes in your environment.</td>
</tr>
<tr>
<td><strong>Performance Counter Date Range Comparison</strong></td>
<td>The Performance Counter Date Range Comparison report provides both graphical and textual data for the selected performance counter(s) over the specified time range. Various parameters can be configured including: Server, Instance, Category, Counter, and Instance. This report is similar to the Performance Counter History report except two different time ranges can be selected.</td>
</tr>
<tr>
<td><strong>Performance Counter History</strong></td>
<td>The Performance Counter History report provides both graphical and textual data for the selected performance counter(s) over the specified time range. Various parameters include: Server, Connection, Category, Counter,</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>and Instance can be configured.</td>
<td></td>
</tr>
<tr>
<td><strong>Server Health</strong></td>
<td>The <strong>Server Health - By Event Severity</strong> report illustrates your current Server Health Score which is based on the severity of active Advisory Conditions in your environment.</td>
</tr>
<tr>
<td>- <strong>By Event Severity</strong></td>
<td>The <strong>Server Health - By Resource Category</strong> report displays your average wait times organized by different resource types.</td>
</tr>
<tr>
<td>- <strong>By Resource Category</strong></td>
<td>Both reports also feature a listing of the Advisory Conditions that were active during that time period and also a bar that graphically represents the amount of uptime that the server has sustained.</td>
</tr>
<tr>
<td><strong>SQL Server Performance</strong></td>
<td>The <strong>SQL Server Performance</strong> report provides detailed SQL Server performance data for the instance over the selected time range. Use the <strong>SQL Server Performance</strong> report in conjunction with the <strong>Performance Analysis Dashboard</strong> to gain an understanding of SQL Server performance in your monitored environment.</td>
</tr>
<tr>
<td><strong>SQL Server Transactions - Top Databases</strong></td>
<td>The <strong>SQL Server Transactions - Top Databases</strong> report provides detailed data about transactions and log flushes for the selected Instance over the specified time range. The report includes a minimum, maximum, average, and total column for each metric. The report includes the top 10 databases for the Instance as determined by the total number of log flushes and transactions.</td>
</tr>
<tr>
<td><strong>SQL Server Wait Stats Analysis</strong></td>
<td>The <strong>SQL Server Wait Stats Analysis</strong> report provides detailed data about SQL Server Waits for the instance over the selected time range. Several aggregate waits groupings are incorporated into the report, including waits by category, by class, and by type. Each grouping breaks down the Waits data by Resource Waits, CPU Waits, and Total Waits. Metrics are displayed in milliseconds (MS).</td>
</tr>
<tr>
<td><strong>SQL Server Wait Stats Analysis - By Resource and CPU</strong></td>
<td>The <strong>SQL Server Wait Stats Analysis - By Resource and CPU</strong> Waits report provides detailed data about SQL Server Waits for the instance over the selected time range. Several aggregate waits groupings are incorporated into the report, including waits by category, by class (CPU or Resource), and by type (CPU or Resource). Metrics are displayed in milliseconds (MS).</td>
</tr>
<tr>
<td><strong>SSAS Performance</strong></td>
<td>The <strong>SSAS Performance</strong> report provides detailed graphical information regarding the SSAS performance for the selected Instance over a given time range.</td>
</tr>
<tr>
<td><strong>Windows + SQL Server Performance</strong></td>
<td>The <strong>Windows + SQL Server Performance</strong> report provides detailed performance metrics for each instance on the selected server with regard to Windows and SQL Server performance. Use the <strong>Windows + SQL Server Performance</strong> report in conjunction with the <strong>Performance Analysis Dashboard</strong> to gain an insight into the performance of your monitored environment.</td>
</tr>
</tbody>
</table>
The **Windows Performance** report provides detailed Windows performance data for the instance over the selected time range. Use the **Windows Performance** report in conjunction with the **Performance Analysis Dashboard** to gain an understanding of Windows performance in your monitored environment.

### BLOCKING

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocking</td>
<td>The <strong>Blocking</strong> reports provide a summary of blocking data over the given time range. They are designed to aggregate by blocking query, blocked query, and wait resource for the selected time frame. Use this data along with other blocking-level metrics available in the <strong>Blocking SQL tab</strong> of Performance Analysis to identify and target problem areas.</td>
</tr>
</tbody>
</table>

### TOP SQL

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top SQL</td>
<td>The <strong>Top SQL</strong> reports provide a summary of Top SQL data captured over the given time range. Top SQL contains your high impact, worst performing stored procedures, batches, and statements. The information captured by Top SQL can be based on any combination of duration, cpu, reads, or writes. Use this data along with other metrics available in the <strong>Top SQL tab</strong> of Performance Analysis to identify and target problem areas. Each of the Top SQL reports will group the Top SQL data by its respective category name.</td>
</tr>
</tbody>
</table>

### DISK / FILE SPACE

Use the Disk/File Space reports in conjunction with the **Disk Activity** and **Disk Space** tabs of **Performance Analysis** to identify data disk capacity issues and possible performance bottlenecks.
<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server File Growth - Top Files - All</td>
<td>The <strong>SQL Server File Growth - Top Files - All Servers</strong> report provides data and log file growth over the selected time range for the Top N files, as selected in the report parameters, across your entire installation. There are representations for <em>Total Size</em> and <em>Used size</em> over the selected time frame.</td>
</tr>
<tr>
<td>SQL Server File Utilization - Top Files</td>
<td>The <strong>SQL Server File Utilization</strong> report provides data and log file growth over the selected time range for the Top 5 files, on the specified Instance. There are representations for <em>Total Size</em> and <em>Used size</em> over the selected time frame. Each of the SQL Server File Utilization reports will group the data by its respective category name.</td>
</tr>
<tr>
<td>- By Logical Disk</td>
<td></td>
</tr>
<tr>
<td>- By Database</td>
<td></td>
</tr>
<tr>
<td>Windows Disk Utilization - By Server</td>
<td>The <strong>Windows Disk Utilization - By Server</strong> report provides the amount of used and free space for the selected server over the specified time range. The report provides data for both Physical and Logical disk utilization.</td>
</tr>
<tr>
<td>Windows Disk Utilization - All Servers</td>
<td>The <strong>Windows Disk Utilization - All Servers</strong> report provides the disk size, used space, free space, and used space history for all of the monitored servers within your SentryOne Environment.</td>
</tr>
<tr>
<td>Misaligned Partitions - By Server</td>
<td>The <strong>Misaligned Partitions</strong> reports provides a summary of the misaligned partitions for the selected server.</td>
</tr>
<tr>
<td>- All Servers</td>
<td></td>
</tr>
<tr>
<td>High VLFs - By Server</td>
<td>The <strong>High VLFs</strong> reports provides data about those transaction logs which have a high number of Virtual Log Files (VLFs).</td>
</tr>
<tr>
<td>- All Servers</td>
<td></td>
</tr>
<tr>
<td>Databases without Recent Backups</td>
<td>The <strong>Databases without Recent Backups</strong> report provides information about Databases without a recent backup in your monitored environment. Information is included for both Data and Log files, including the Last Backup Time, Last Backup Type, and Used Size (MB). The default time range for this report covers the last 15 days, but is configurable. Transaction Log files are not included in this report if the database is in simple recovery.</td>
</tr>
</tbody>
</table>
### Deadlocks

<table>
<thead>
<tr>
<th>Description</th>
<th>By Application</th>
<th>By Application Matrix</th>
<th>By Resource</th>
<th>By SQL Server</th>
<th>By User</th>
</tr>
</thead>
</table>

The Deadlocks reports provide information regarding deadlocks in various ways over a specified time range. When used in conjunction with the Deadlocks tab in Performance Analysis, these reports can help track down problematic deadlocks.

### GENERAL

#### Action Counts

The Action Counts report provides a count, broken down by object and parent object, of Actions that have taken place during the selected time range. This report can give you valuable information for fine tuning alerts in your environment. Use the Action Counts report to quickly identify where the alerts in your environment are coming from and then make any adjustments as needed.

#### Active Settings List

The Active Settings List report provides details on every setting configured in your environment at all levels combined into one resource. This report can give you valuable information about how settings are effecting your configured actions and collection settings.

#### Configured Actions List

The Configured Actions List report provides a list on every action that has been configured for a given condition, specifying at what level the action has been configured.

#### Configured Notification List

The Configured Notification List report allows you to view all conditions that are specifically configured to send notifications by email or pager.

#### Downtime - By Server

The Downtime - By Server report provides a listing of all server downtime activity for the specified time range. Included in the reports are metrics such as total incidents, total downtime in seconds, average downtime per incident, uptime percentage, and first/last incident times.

### REPORT DEPLOYMENT

You can deploy any SentryOne report to one or more Reporting Services 2005 and above servers. Deploying reports to your own SSRS instance allows you to create custom deployment schedules. Before the reports can be deployed, you must specify the name of the server where you wish to deploy the reports.

### DEPLOYMENT SETTINGS
The Report Deployment Settings may be accessed from the Reports menu (Reports → Deployment Settings). The following Reporting Service endpoints are supported for deploying SentryOne Reports:

SSRS 2005/2008 native mode:


SSRS 2008 R2 or higher, native mode mode:


When you choose to deploy your reports, you will be given the ability to decide where on your reporting server that you would like to store these reports. Simply click the desired folder and press OK. You may create new folders by right-clicking on an existing folder and choosing New Folder.

Note: Windows Authentication must be enabled to immediately run reports via the web interface.

REPORT SUBSCRIPTIONS

In order to create a subscription for any of the SentryOne Reports you will need to adjust the Instance properties of the SQL Data Source SQLDS as follows. In the Properties of the data source change the Connect using settings to the Credentials stored securely in the report server option. Beneath this, be sure to also select the Use as Windows credentials when connecting to the data source option.

ADDITIONAL REPORTING FEATURES

<table>
<thead>
<tr>
<th>List Views</th>
<th>All list views inside the SentryOne Client can be easily exported.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime Stats</td>
<td>All runtime stats in the SentryOne Client are printable.</td>
</tr>
<tr>
<td>Performance Graphs</td>
<td>All performance graphs in the SentryOne Client are printable.</td>
</tr>
<tr>
<td>Event Chains</td>
<td>All Event Chain diagrams in the SentryOne Client are printable.</td>
</tr>
<tr>
<td>Calendar Views</td>
<td>All Calendar views in the SentryOne Client are printable.</td>
</tr>
</tbody>
</table>

IMPORT REPORTS

This option can be found in the Reports menu. SentryOne supports importing future reports (created by the SentryOne development team) to the Client separately without requiring an upgrade.

6.3.2 Adding Notes

Notes are used to help facilitate communication between DBA team members. They can be used for change tracking, resolution history, and a variety of other functions. You can set notes to be included in any future notifications generated for an object, as well as to be displayed for all
calendar instances of an object. Both of these features can be valuable for sharing escalation procedures, recovery steps, emergency contact information, etc., with other team members.

The Notes interface can be reached through a variety of context menus throughout the SentryOne Client, just about anywhere that you can act on an event object or event instance. The notes interface can also be accessed by pressing (CTRL + ALT + N) when an applicable object is selected.

When adding a note, if any failures exist for the selected event object, you will be prompted to clear them.

<table>
<thead>
<tr>
<th>Field</th>
<th>Hotkey</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>n/a</td>
<td>Displays the current Windows user name.</td>
</tr>
<tr>
<td>Title</td>
<td>n/a</td>
<td>The subject of the note which will appear in the selection list.</td>
</tr>
<tr>
<td>Note</td>
<td>n/a</td>
<td>The body of the note which cannot be blank.</td>
</tr>
<tr>
<td>Include in Notifications</td>
<td>n/a</td>
<td>Enable this setting to include this note with any notifications related to the object.</td>
</tr>
<tr>
<td>Show for All Calendar Instances</td>
<td>n/a</td>
<td>Enable this setting to show the note information in the popup of every event instance of the object.</td>
</tr>
<tr>
<td>New</td>
<td>Ctrl+N</td>
<td>Creates a new note.</td>
</tr>
<tr>
<td>Save</td>
<td>Ctrl+S</td>
<td>Saves the current note.</td>
</tr>
<tr>
<td>Delete</td>
<td>Ctrl+D</td>
<td>To delete a note, select it from the list and click delete.</td>
</tr>
<tr>
<td>Close</td>
<td>Ctrl+C</td>
<td>Closes the window.</td>
</tr>
</tbody>
</table>
6.3.3 Monitor Additional Targets and Instances

**TERMINOLOGY**

When the word *Target* is used, we are referring to the device that houses your data, whether it's a physical server, cloud installation, or APS appliance.

*Instance* is referring to an instance of SQL Server or SSAS that exists on a Target in your environment.

**SUPPORTED TARGETS AND INSTANCES**

Currently, SentryOne supports the monitoring of Windows, SQL DW, Azure SQL Database, VMware Host, and APS appliance Targets. Supported Instances include SQL Server and SSAS. For additional details, including supported versions, please see our Quick Start guide.

**ACCESS LEVEL**

When adding a new Target, the first step is a Feature Availability test. The results of this test will determine whether the Target will be added with Full Access or Limited Access. When a Target is added with Full Access, the monitoring service will collect Windows level metrics and you will have full access to the features of Performance Analysis.

If the Target fails the Feature Availability test, you can click the Troubleshooting link and attempt to resolve the issue. After applying a solution, you can then retest the Target. There are some situations in which Limited Access is the only option. For example, if you are monitoring a cloud-based SQL Server instance, you will likely not have access to the OS. When Limited Access is applied, the monitoring service will not collect Performance Counters, and access to the Windows Dashboard, Disk Activity tab, Disk Space tab, and Windows Processes tab will be restricted.

**ADDING INSTANCES**

You can easily add additional monitored Instances to your SentryOne environment. This is accomplished by right-clicking either the All Targets node, a Site node, a Target Group node, or an existing Target node in the Navigator pane and using the *Add Instance* command. You can also add an Instance through the *File* menu.

In the *Add Instance dialog* you may choose the desired *Instance Type* from the drop-down menu (Analysis Services Instance, SQL Server Instance).

**WATCHING INSTANCES**

Before SentryOne will start monitoring an Instance or object, its status must be set to “watched”. Instances or objects that are not being watched will be displayed with a grayed-out icon next to their name in the Navigator tree view.
Unwatched Instances or objects can have their status set to watched through their respective context menus with the **Watch** command. Once you have watched a new Instance the SentryOne Monitoring Service will start actively monitoring the Instance and its objects, and begin honoring any associated configured Conditions and Actions.

**Note:** Immediately after adding an Instance or setting an Instance to watched status SentryOne will begin to synchronize with that Instance. Exactly how long the synchronization process takes depends on the number of objects associated with the Instance, the amount of historical data available, and how many Instances are being watched at the same time. The *Watch Status Window* will keep you informed of the process and alert you about any errors.

**MODIFYING INSTANCE PROPERTIES**

After you’ve added an Instance, you may need to change how SentryOne connects to the target. When right clicking on an Instance, you’ll see two options for connection properties: User and Monitoring Service.

**User Connection Properties** define how your SentryOne Client will connect to a monitored server for the current user. These properties can vary for each Client in your environment. The SentryOne Client only connects directly to a monitored server under specific scenarios. More details regarding those scenarios and specific security requirements can be found in the *Client Security* section of our Quick Start Guide.

**Monitoring Service Connection Properties** define how the SentryOne Monitoring Service will connect to the selected server. The setting can be applied from any SentryOne Client. The Monitoring Service Connection can be configured by right clicking on the Instance and selecting Monitoring Service Connection Properties from the context menu. Please see our Quick Start Guide for additional information on Monitoring Service Security.

Within the Connection Properties window, there are several properties that can be changed.

- Enable Integrated Authentication. This setting tells the Instance to use the integrated Windows account information.
- Credentials. This is where you enter SQL Server credentials if you are not using Integrated Authentication.
- Alias. By default, you will see the server name that you initially entered when adding the Instance.
- Port. This setting is used to connect to SQL Server if it has been configured to a non-standard port.
- Access Level. This setting is used to assign the level of access that SentryOne has to the selected target. A Target with limited access will not be able to access Windows based features, such as the Windows Dashboard, Windows Processes tab, Disk Space tab, or Disk Activity tab. Limited access targets will also not have access to PerfLib Performance Counters for that target.

All of these settings are available for SQL Server Instances. SSAS Instances only offer the Port setting. Targets offer the Access Level setting.
6.3.4 Connect to Installation

The **Connect to Installation**... File menu item controls the active database for the locally installed SentryOne Monitoring Service and Client.

**Please Specify the Server**

Enter the name of the Server to connect to in the field provided.

**Please Specify the Database Name**

Specify the database name of the SentryOne Database that you wish to connect to.

**Use Integrated Windows Authentication**

By Default this option will be selected. If you wish to use SQL Server authentication, uncheck the box and enter the SQL Server credentials. After connecting to the server a new instance of the SentryOne Client will be launched that connects to the specified alternate installation.

**To Remove an old Server from the All Instances View**

When you start the SentryOne Client, if you have more than one server in the All Instances View you will be prompted to choose an installation to connect to. You may have more than one server in the All Instances View if you have moved the SentryOne Database to a new location.

To remove an old server from the All Instances View so that you will no longer be prompted about it when opening the SentryOne Client:
1. Select the *Server Name* of the server you would like to remove in the “Please Specify the Server Name” drop-down list, and press Shift + Delete.

2. The new server should now be the only one listed in the drop-down list. Use the Connect command and connect to the new installation.

The next time you open the Client it will automatically connect to the new installation.
SentryOne employs a comprehensive Alerting and Response System. The Alerting and Response System uses the concept of **Conditions** and **Actions**. Conditions describe the various states of monitored objects. SentryOne allows you to define criteria for when **Conditions** are met using various **Settings** including **Condition Settings**. As a response to **Conditions** taking place **Actions** can be configured. You can refine how often **Actions** will be carried out using **Response Rulesets**. To control the time frame of when an **Action** will be processed you may also apply a **Window** directly to any configured Action.

**Conditions** and **Settings** are all hierarchical, working through the principle of inheritance. This means that Conditions and Settings applied at the global level (**All Targets node**), are automatically inherited by all objects below that node. Any Inherited Conditions or Settings can then be overridden at each lower level as needed.

In practice, this allows you to design your alerts and responses at a high level (Global All Targets) and then fine tune them as necessary at the lower levels (Sites, Targets, Target Groups, Instances, Objects). For a visual representation of these different levels see the **Alerting and Response System Hierarchy diagram**.

The following table contains a brief description of the different concepts within the SentryOne Alerting and Response system. For detailed information concerning each concept, select the links to open the individual sub-topics.

**Concepts:**

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ <strong>Contact Management</strong></td>
<td>The <strong>Contacts</strong> Navigator node is used to create and maintain Users and Groups within your SentryOne environment. Once a User or Group has been created they can be chosen as a target for the Send Email /Send Page Action.</td>
</tr>
<tr>
<td>→ <strong>Conditions</strong></td>
<td><strong>Conditions</strong> describe the various states of monitored event objects or associated performance counters ('Schedule Performance Monitoring' in the on-line documentation). You define <strong>Actions</strong> to take when <strong>Conditions</strong> are met.</td>
</tr>
<tr>
<td>→ <strong>Condition Settings</strong></td>
<td><strong>Condition Settings</strong> allow you to define rules that a <strong>Condition</strong> must meet in order for the <strong>Condition</strong> to be fully satisfied. <strong>Condition Settings</strong> are applicable to most General Conditions and all Audit and Failsafe conditions and are configured using a visual filter editor.</td>
</tr>
<tr>
<td>→ <strong>Actions</strong></td>
<td><strong>Actions</strong> determine what happens when a <strong>Condition</strong> is met, including sending notifications, executing a process, etc.</td>
</tr>
<tr>
<td>→ <strong>Settings</strong></td>
<td><strong>Settings</strong> define criteria for when a <strong>Condition</strong> is considered to be met, including runtime thresholds for events that are captured. Certain <strong>Settings</strong>, known as <strong>Source Settings</strong> define what events are collected by SentryOne.</td>
</tr>
<tr>
<td>→ <strong>Response Rulesets</strong></td>
<td><strong>Response Rulesets</strong> control how often <strong>Actions</strong> are taken in</td>
</tr>
</tbody>
</table>
response to **Conditions** being met. They do this by assigning additional criteria on top of the **Condition** itself, that must be met before an **Action** will take place. The default **Response Ruleset** is **Notify EveryTime**.

**Windows**

*Windows* may be applied directly to any configured **Action** to control the time frame of when that **Action** will be processed. Windows may also be applied to users or groups.

**Object Groups**

*Object Groups* exist outside of the normal SentryOne Hierarchy; an **Action** or **Setting** configured for an Object Group will be applied last, after any inherited or explicitly defined Action or Setting. *Object Groups* are useful when you need to apply like policies to a set of objects that do not exist within the same hierarchical groups in your SentryOne environment.

### ALERTING AND RESPONSE SYSTEM HIERARCHY

**Actions** and **Settings** have a hierarchical configuration. This means that settings applied at the Global level (**All Targets**) are automatically inherited by all objects below that node. Inherited Conditions and Settings can then be overridden at each level as needed. These new **Conditions** and **Settings** will then be inherited by each object below that level. The below diagram shows the flow of inherited **Conditions** and **Settings**. It also shows those levels at which the Conditions and Settings panes are available for configuration.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conditions</strong></td>
<td>describe the various states of an <em>event object</em> or associated <em>performance counters</em> ('Schedule Performance Monitoring' in the on-line documentation). You can configure <strong>Actions</strong> to take place when a Condition is met. All <strong>Actions</strong> work on the principle of inheritance. This means that if you configure an <strong>Action</strong> in response to a <strong>Condition</strong> being met at the Global level (<strong>Shared Groups</strong>), it will be automatically passed down to all applicable objects below it. This allows the user to define Global <strong>Actions</strong> for the most common issues across your environment once, and have those passed down to every monitored server automatically. You can further refine <strong>Actions</strong> at each level as needed.</td>
</tr>
</tbody>
</table>
For a general overview of the Alerting and Response System see the [Alerting and Response System](#) topic. For a visual representation of how inheritance works within SentryOne see the [Alerting and Response System Hierarchy diagram](#).

Conditions fall into four categories: **General, Failsafe, Audit, and Advisory**.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Conditions</strong></td>
<td>General Conditions are applicable to monitored event objects or associated performance counters. These are the most common Conditions which you will configure Actions for in your environment. Not all General Conditions apply to all event object types. Condition Settings are available for most General Conditions. Condition Settings allow you to define additional rules that a Condition must meet in order for the Condition to be fully satisfied. Example General Conditions include SQL Agent job failure, Event Chain completed, and Runtime Threshold conditions. For a complete list of General Conditions see the General Conditions topic. You can configure Actions in response to General Conditions being met at the Global level (Shared Groups), at the Site level, at the Target level, at the Instance level, and at the individual object level.</td>
</tr>
<tr>
<td><strong>Failsafe Conditions</strong></td>
<td>Failsafe Conditions fall into two categories: Instance Status and Error. Instance Status Failsafe Conditions are met when an Instance goes Offline or Online. Error Failsafe Conditions are met when there is an error detected with a SentryOne Monitoring Service or related monitoring process. Example Failsafe Conditions falling under the Instance Status category include SQL Server Agent Offline and Analysis Server Offline. Example Failsafe Conditions of the Error category include Monitoring Service errors and Reporting Services monitor errors. For a complete list of all Failsafe Conditions see the Failsafe Conditions topic. You can configure Actions in response to Instance Status Conditions being met, at the Global level (Shared Groups), at the Site level, at the Target level, or at the individual Instance level. You can configure Actions in response to Error Conditions being met at the Global level (Shared Groups) or at the Site level.</td>
</tr>
<tr>
<td><strong>Audit Conditions</strong></td>
<td>Audit Conditions are applicable to various activities carried out both inside and outside of the SentryOne Client. Audit Conditions provide a way for you to be notified when various Settings are changed, or monitored objects are modified from within the SentryOne Client. Example Audit Conditions include Settings changed, Actions changed, or Response Rulesets changed. For a complete list of all Audit Conditions see the Audit Conditions topic. You can configure Actions in response to Audit Conditions being met</td>
</tr>
</tbody>
</table>
at the Global level (Shared Groups), at the Site level, at the Target level, or at the individual Instance level.

Advisory Conditions allow for the creation of conditions that are customized to your environment as you see fit.

Use the links below to jump to a description of the various Conditions. For each Condition you will find a description and any related Setting(s) which may affect the Condition.

Jump To:

General Conditions
- Blocking SQL
- Deadlocks
- Event Chain
- Event Chain Node
- Index
- Maintenance Plan
- Performance Counter
- Reporting Services Report
- SQL Server Agent Alert
- SQL Server Agent Job
- SQL Server Agent Log
- Top Commands
- Top SQL
- Windows Instance

Failsafe Conditions
- Analysis Services
- Monitoring Service
- SQL Server Instance
- Windows Instance

Audit Conditions
- Custom Event View
- Event Chain
- Global
- Performance Counter

7.1.1 Condition Settings
Condition Settings allow you to define rules that a Condition must meet in order for the Condition to be fully satisfied. Conditions describe the various states of monitored objects within your environment. SentryOne allows you to define Actions to be taken when Conditions are met. Through the use of advanced filtering, the Condition Settings tab gives you complete control of when an Action should be taken for a given Condition.

Condition Settings allow you to define any number of criteria surrounding captured events. The criteria you define must be met in order for the Condition to be satisfied. Condition Settings can be accessed in the Condition Settings tab of any configured Condition/Action combination. Logical operators may be used to form complex conditional groups, allowing you to define criteria across any number of columns.

Whenever you configure a Condition you also specify an Action to take place when that Condition is met. If an event does not meet the Condition Settings criteria, the Condition is not considered to be met, and the associated Action will not be taken.

CONFIGURING CONDITION SETTINGS

Condition Settings are available for most General Conditions and all Audit and Failsafe Conditions. Condition Settings can be accessed from the General Conditions pane for configured Condition/Action combinations. The Condition Settings tab contains both a Visual Filter Editor and a Text Filter Editor.

Please see the Filter Editor topic for details about the Filter Editor, including context menu information, and example uses for each criteria operator.

Message Text

Message Text is a useful column to configure a Condition Settings filter on. Message Text corresponds to the entire body of an Alert notification received for each respective Condition. Any text you see in the body of alert emails will be evaluated as part of the Message Text filter.

Event Text

The Event Text filter is available for select Conditions. Event Text most commonly corresponds to the raw output text of an event. For Deadlock events this will filter on the captured raw xml of the Deadlock. For Defragmentation events the filter will be applied to the last operation text, which may contain events such as a successful offline rebuild or error information that happened during the operation.

CONDITION SETTINGS EXAMPLES

The following section contains example walkthroughs for configuring different filters within the Condition Settings tab. Click on a heading to expand the section.

Index Defragmentation Completed Example

INDEX DEFRAGMENTATION COMPLETED EXAMPLE

Fragmentation Manager can perform Reorganization or Rebuild operations based on fragmentation thresholds you specify. You may wish to be notified if the fragmentation level of an index is still above a certain threshold after a Defragmentation Operation has completed.
In this example we will configure an email alert for the **Index: Defragmentation Completed Condition**, which will only fire after a Defrag operation has taken place and the fragmentation level remains above 15%.

> Keep in mind that there are a number of Actions which can be taken in response to Conditions being met. You could always choose to execute an Agent Job or a TSQL command, either in conjunction with, or instead of sending an alert email.

**Add the Condition**

1. Select the Shared Groups node in the Navigator pane to set the Action globally. Open the General Conditions pane (View menu → General Actions).
2. In the General Conditions pane click the Add button to open the Actions Selector.
3. Expand the Index group and then the **Index: Defragmentation Completed Condition**. Select the check box next to the Send Email Action and then the OK button.
4. In the Actions Settings tab select your desired user or group to target with the Send Email Action.

The configured **Send Email Action** for the **Index: Defragmentation Completed Condition** is now listed in the General Conditions pane.

**Add a Condition Setting to the Condition**

1. Select the Condition Settings tab in the General Conditions pane. The Condition Filter is initially empty.
2. Next to the And logical Operator, you will see a plus symbol. Select the plus symbol to add a new Condition Filter.
3. Select [Table Name] to open a drop-down list of available choices and choose Current Fragmentation Level → Value [Current Fragmentation Level.Value].
4. For the Criteria Operator open the drop-down list and choose Is greater than.
5. Select the <enter a value> field and enter your desired value. For this example we will enter 15.

The **Condition Setting** is now configured. If you select the Condition Settings Text tab you should see the filter string defined as follows:

\[
\text{[Current Fragmentation Level.Value]} > 15.0
\]

With the **Condition Setting** defined as above, the **Index: Defragmentation Completed Condition** will be met, and its associated **Send Email Action** will fire for the following case:

- A Defragmentation Operation has completed on an index and the fragmentation level is above 15%.
SQL Server Deadlock Example

SQL SERVER DEADLOCK EXAMPLE

SentryOne collects information about deadlocks happening within your environment. You can view this information in the Deadlocks tab of Performance Analysis.

Depending on your environment you may not wish to be alerted about every deadlock which takes place. In this example we will configure an alert for the SQL Server: Deadlock Condition and limit the notifications you receive to a specific database using Condition Settings.

Add the Condition

1. Select the Shared Groups node in the Navigator pane to set the Action globally. Open the General Conditions pane (View menu → General Actions).
2. In the General Conditions pane click the Add button to open the Actions Selector.
3. Expand the Index group and then the SQL Server: Deadlock Condition. Select the check box next to the Send Email Action and then the OK button.
4. In the Actions Settings tab select your desired user or group to target with the Send Email Action.

Add a Condition Setting to the Condition

1. Select the Condition Settings tab directly beneath the Explicit section of the General Conditions pane. The Condition Filter is initially empty.
2. Next to the And logical Operator, you will see a plus symbol. Select the plus symbol to add a new Condition Filter.
3. Select [Application Name] to open a drop-down list of available choices and choose [Database Name].
4. For the Criteria Operator open the drop-down list and choose Equals.
5. Select the <enter a value> field and enter your desired value. For this example we will enter 'AdventureWorks'.

The Condition Setting is now configured. If you select the Condition Settings Text tab you should see the filter string defined as follows:

[Database Name] = 'AdventureWorks'

With the Condition Setting defined as above, the SQL Server: Deadlock Condition will be met, and its associated Send Email Action will fire for the following case:

- A Deadlock is detected and the victim database is the AdventureWorks.
7.1.1.1 Filter Editor

SentryOne allows you to build complex filters for the purposes of History Filtering and defining Condition Settings. This topic covers the functionality of the Filter Editor, which is common to both History Filtering and Condition Settings.

Note: The Filter Editor is case-insensitive for all string comparisons.

FILTER EDITOR OVERVIEW

The below diagram identifies the various aspects of the Filter Editor.

FILTER RULES

Using the Filter Editor, you build criteria that an event must meet by defining rules.

Each filter rule is made up of three distinct parts.

1. First, you define the scope of the rule, by specifying a column name.
2. Secondly, you select an operator (criteria operator) which defines an instruction for testing the value (operand value).
3. Finally, you enter the value (operand value) for this filter rule.

RULE GROUPS

Individual rules are logically grouped into rule groups. Each group contains a logical operator which defines how the rules are combined within the group. Upon creation each group will use the And logical operator. When a group is created with the And logical operator each filter rule that is part of the group must be true in order for the group to evaluate as true. You may change the logical operator defined for a group by first clicking it; a drop down menu will appear, allowing you to choose an alternative (And, Or, Not And, Not Or).

CONTEXT MENUS

When you click on any logical operator additional options are available from the drop-down
menu.

- The **Add Condition** command will add a new filter rule to the condition group.
- The **Add Group** command will create a new sub-group beneath the existing group with the default And logical operator.
- The **Remove Group** command will delete the selected group.
- The **Clear All** command, available from the root logical operator, will delete all filter rules.

Selecting the plus symbol next to any group's logical operator will add a new filter rule to the group. Selecting the symbol next to any filter rule will delete that rule from the group.

### 7.1.2 General Conditions

General Conditions are applicable to monitored event objects or associated performance counters (‘Schedule Performance Monitoring’ in the on-line documentation). These are the most common Conditions which you will configure Actions for within your environment. Please keep in mind that not all General Conditions apply to all event object types. A Condition must be supported by an object's underlying event provider in order for Conditions and Actions to be available in SentryOne.

**CONFIGURE ACTIONS FOR GENERAL CONDITIONS**

**Configure Actions for General Conditions**

You can configure Actions in response to General Conditions being met at these levels: Shared Groups (Global), Site, Target Group, Target, and Instance.

To do so, select the node appropriate to the level you would like to configure the action for in the Navigator Pane and then the General Actions tab in the Conditions pane. If you do not see the Conditions pane use the View menu → General Actions.

Next you will want to click the Add button found in the Conditions pane. This will open the Select Action window. Expand the applicable Object and Condition. Use the check box(s) to select which Actions should be taken in response to this Condition being met. Click the OK button.

For more information about Actions that can be taken when a Condition is met see the Actions topic.

**COMMON GENERAL CONDITIONS**

**Common General Conditions**

The Following table contains a description of several common General Conditions.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Is Met When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Started</td>
<td>Occurs when an event instance begins.</td>
</tr>
<tr>
<td>Completed</td>
<td>Occurs when an event instance finishes running, regardless of success or failure status.</td>
</tr>
<tr>
<td>Success</td>
<td>Occurs when an event instance completes and returns a success status.</td>
</tr>
<tr>
<td>Failure</td>
<td>Occurs when an event instance completes and returns a failure status.</td>
</tr>
<tr>
<td>Output Content Match</td>
<td>This condition is met when specific content, which you define, exists in the output message text for an event instance. See the Output Content Match topic for more information.</td>
</tr>
</tbody>
</table>
Conditions | Is Met When
---|---
Runtime Threshold Min | Occurs whenever an event runs shorter than x% of its average runtime, or shorter than an explicit duration.
Runtime Threshold Max | Occurs whenever an event runs longer than x% of its average runtime, or longer than an explicit duration.
Performance Counter Threshold Min | Occurs when a minimum performance threshold has been set for a performance counter, and the counter’s value falls below it.
Performance Counter Threshold Max | Occurs when a maximum performance threshold has been set for a performance counter, and the counter’s value exceeds it.

**GENERAL CONDITIONS COMPREHENSIVE LIST**

Use the links below to jump to a description of the various Conditions. For each Condition, you will also find a column which lists related Settings which affect the Condition. Settings define criteria for when a Condition is considered to be met, including runtime thresholds for events that are captured. Certain Settings, known as Source Settings define what events are collected by SentryOne. For more information about Settings see the Settings topic.

**BLOCKING SQL**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocking SQL</td>
<td>A Block was detected, subject to the Minimum Block Duration set for the Blocking SQL Source. View the block on the Event Calendar or the Performance Analysis Blocking SQL tab.</td>
<td>Blocking SQL Source: Minimum Block Duration</td>
</tr>
<tr>
<td>Blocking SQL: Output Content Match</td>
<td>A match condition was found in the output content of a Block.</td>
<td>Blocking SQL Source: Minimum Block Duration</td>
</tr>
<tr>
<td>Blocking SQL: DurationThreshold Max</td>
<td>Block exceeded the maximum duration threshold.</td>
<td>Blocking SQL: Maximum Duration Threshold</td>
</tr>
</tbody>
</table>

**DEADLOCKS**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadlock</td>
<td>A Deadlock was detected. View the deadlock on the Event Calendar or the Performance Analysis Deadlocks tab.</td>
<td>Deadlocks Source: Collect Deadlock Events</td>
</tr>
</tbody>
</table>
## Deadlock: Output Content Match

A match condition was found in the output content of a Block. You must define which strings to search for in the event output to trigger this condition. See Condition Settings for more information.

### EVENT CHAIN

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>Event Chain execution completed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Failure</td>
<td>Event Chain execution failed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Runtime Threshold Max</td>
<td>Event Chain exceeded the maximum runtime threshold.</td>
<td>Event Chain: Maximum Runtime Threshold</td>
</tr>
<tr>
<td>Runtime Threshold Min</td>
<td>Event Chain did not meet the minimum runtime threshold.</td>
<td>Event Chain: Minimum Runtime Threshold</td>
</tr>
<tr>
<td>Started</td>
<td>Event Chain execution started.</td>
<td>N/A</td>
</tr>
<tr>
<td>Success</td>
<td>Event Chain execution succeeded.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### EVENT CHAIN NODE

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>Event Chain Node execution completed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Failure</td>
<td>Event Chain Node execution failed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Runtime Threshold Max</td>
<td>The runtime threshold maximum for an Event Chain Node was exceeded.</td>
<td>Event Chain Node: Maximum Runtime Threshold</td>
</tr>
<tr>
<td>Runtime Threshold Min</td>
<td>Event Chain Node did not meet the minimum runtime threshold.</td>
<td>Event Chain Node: Minimum Runtime Threshold</td>
</tr>
<tr>
<td>Started</td>
<td>Event Chain Node execution started.</td>
<td>N/A</td>
</tr>
<tr>
<td>Success</td>
<td>Event Chain Node execution succeeded.</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### INDEX

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defragmentation Complete</td>
<td>Index Defragmentation Completed.</td>
<td>Index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defragmentation</td>
</tr>
<tr>
<td>Defragmentation Failure</td>
<td>Index Defragmentation Failed.</td>
<td>Index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defragmentation</td>
</tr>
<tr>
<td>Defragmentation Started</td>
<td>Index Defragmentation has started.</td>
<td>Index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defragmentation</td>
</tr>
</tbody>
</table>

### MAINTENANCE PLAN

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Content Match</td>
<td>A match condition was found in the output content of a Maintenance Plan.</td>
<td>Condition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Settings</td>
</tr>
<tr>
<td>Step Failure</td>
<td>Maintenance Plan step failed.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### PERFORMANCE COUNTER

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold Max</td>
<td>Performance Counter value exceeded the maximum threshold.</td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thresholds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum</td>
</tr>
<tr>
<td>Threshold Min</td>
<td>Performance Counter did not meet the minimum threshold</td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thresholds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum</td>
</tr>
</tbody>
</table>

### REPORTING SERVICES REPORT

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>Reporting Services Report completed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Failure</td>
<td>Reporting Services Report failed.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Reporting Services Report exceeded the maximum runtime threshold.</td>
<td>Reporting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Services Report</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Runtime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Threshold</td>
</tr>
<tr>
<td>Started</td>
<td>Reporting Services Report started.</td>
<td>N/A</td>
</tr>
<tr>
<td>Success</td>
<td>Reporting Services Report succeeded.</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### SQL SERVER AGENT ALERT

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert Fired</td>
<td>SQL Server Agent Alert was fired.</td>
<td>N/A</td>
</tr>
<tr>
<td>Output Content Match</td>
<td>A match condition was found in the output content of a SQL Server Agent Alert.</td>
<td></td>
</tr>
</tbody>
</table>

### SQL SERVER AGENT JOB

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>SQL Server Agent Job is blocked or is blocking.</td>
<td>N/A</td>
</tr>
<tr>
<td>Completed</td>
<td>SQL Server Agent Job completed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Conflict</td>
<td>There was a job conflict, meaning two or more jobs were seen running concurrently.</td>
<td>N/A</td>
</tr>
<tr>
<td>Failure</td>
<td>SQL Server Agent Job failed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Output Content Match</td>
<td>A match condition was found in the output content of a SQL Server Agent Job.</td>
<td></td>
</tr>
<tr>
<td>Queued</td>
<td>SQL Server Agent Job was queued via SentryOne's queuing system.</td>
<td></td>
</tr>
<tr>
<td>Retry</td>
<td>SQL Server Agent Job step was retried.</td>
<td>N/A</td>
</tr>
<tr>
<td>Run Missed</td>
<td>The scheduled run for a job was missed for some reason. This may be because SQL Server Agent was offline.</td>
<td></td>
</tr>
<tr>
<td>Runtime Threshold Max</td>
<td>SQL Server Agent Job exceeded the maximum runtime threshold.</td>
<td></td>
</tr>
</tbody>
</table>

The Minimum Block Duration Required to Trigger "SQL Server Agent Job: Block" Condition can be changed in the [Monitoring Service Setting](#).
<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime Threshold Min</td>
<td>SQL Server Agent Job did not meet the minimum runtime threshold.</td>
<td>SQL Server Agent Job: Minimum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Runtime Threshold</td>
</tr>
<tr>
<td>Started</td>
<td>SQL Server Agent Job started.</td>
<td>N/A</td>
</tr>
<tr>
<td>Step Failure</td>
<td>SQL Server Agent Job step failed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Success</td>
<td>SQL Server Agent Job succeeded.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**SQL SERVER AGENT LOG**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>An Error event occurred in the SQL Server Agent Log.</td>
<td>N/A</td>
</tr>
<tr>
<td>Output Content Match</td>
<td>A match condition was found in the output content of a SQL Server Agent Log.</td>
<td>Condition Settings</td>
</tr>
<tr>
<td>Warning</td>
<td>A Warning event occurred in the SQL Server Agent Log.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**TOP COMMANDS**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Commands: Completed</td>
<td>An Analysis Services command (MDX or XMLA) completed.</td>
<td>Top Commands Source: Collect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDX/DMX/XMLA Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Top Commands Source: Minimum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Duration</td>
</tr>
<tr>
<td>Top Commands: Error</td>
<td>An Analysis Services command (MDX or XMLA) completed with an error.</td>
<td>Top Commands Source: Collect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDX/DMX/XMLA Events</td>
</tr>
<tr>
<td>Top Commands: Output Content Match</td>
<td>A match condition was found in a completed Analysis Services command (MDX or XMLA).</td>
<td>Top Commands Source: Minimum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Duration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Condition Settings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Top Commands: Maximum</td>
</tr>
</tbody>
</table>
### Top Commands

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Commands: Runtime Threshold Max</td>
<td>The runtime threshold maximum was exceeded for an Analysis Services command (MDX or XMLA).</td>
<td>Top Commands Source: Collect MDX/DMX/XMLA Events  Top Commands Source: Minimum Duration</td>
</tr>
</tbody>
</table>

### Top SQL

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top SQL: Completed</td>
<td>Top SQL event completed. NOTE: It is highly recommended that a ruleset be used with this condition to avoid excessive logging.</td>
<td>Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source</td>
</tr>
<tr>
<td>Top SQL: Error</td>
<td>Top SQL event completed with an error.</td>
<td>Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source</td>
</tr>
<tr>
<td>Top SQL: Output Content Match</td>
<td>A match condition was found in the content of a completed Top SQL event.</td>
<td>Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source</td>
</tr>
<tr>
<td>Top SQL: Runtime Threshold Max</td>
<td>Top SQL event exceeded the maximum runtime threshold.</td>
<td>Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source  Top SQL Source</td>
</tr>
</tbody>
</table>

### Windows Task

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
<th>Related Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>Windows Task completed.</td>
<td>N/A</td>
</tr>
<tr>
<td>Failure</td>
<td>Windows Task failed, as determined by the success exit code setting for the task.</td>
<td>N/A</td>
</tr>
<tr>
<td>Output Content Match</td>
<td>A match condition was found in the output content for a Windows Task.</td>
<td>Condition Settings  Windows Task  Maximum Runtime Threshold  Windows Task Maximum Runtime</td>
</tr>
</tbody>
</table>
Threshold

Started
Windows Task started.

Success
Windows Task succeeded, as determined by the success exit code setting for the task.

N/A

N/A

WINDOWS EVENT LOG

An event occurred in a Windows Event Log.

Condition Settings can help to restrict which events you are alerted about.

Condition
Description
Related
Setting

Started
Windows Task started.
N/A

Success
Windows Task succeeded, as determined by the success exit code setting for the task.
N/A

7.1.3 Failsafe Conditions

Failsafe Conditions fall into two categories: Connection Status and Error. Instance Status
Failsafe Conditions are met when an Instance goes Offline or Online. The Offline Condition for an Instance will be met if a server crashes or becomes otherwise unavailable. The Online condition will be met once that Instance becomes available again. Failsafe Error Conditions are met when there is an error detected with a SentryOne Monitoring Service or related monitoring process.

The Log to Database Action and the Send Email Action are both configured by default for all Failsafe Conditions. This is so you can be alerted whenever an Instance goes offline, or whenever the SentryOne Monitoring Service experiences an error and is unable to perform its work.

CONFIGURE ACTIONS FOR FAILSAFE CONDITIONS

Configure Actions for Failsafe Conditions

You can configure Actions in response to Instance Status Failsafe Conditions being met at these levels: Shared Groups (Global), Site, Computer Group, Computer, and Instance. You can configure Actions in response to Failsafe Error Conditions being met at the Global level (Shared Groups) or at the Site level.

To do so, select the node appropriate to the level you would like to configure the action for in the Navigator Pane and then the Failsafe Actions tab in the Conditions pane. If you do not see the Conditions pane use the View menu → Failsafe Actions.

Next you will want to click the Add button found in the Conditions pane. This will open the Select Action window. Expand the applicable Object and Condition. Use the check box(es) to select which Actions should be taken in response to this Condition being met. Click the OK button.

For more information about Actions that can be taken when a Condition is met see the Actions
Below you will find a complete listing of all Failsafe conditions.

## Failsafe Conditions

### Analysis Services

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td>Analysis Services is offline or otherwise inaccessible.</td>
</tr>
<tr>
<td>Online</td>
<td>Analysis Service is online and accessible.</td>
</tr>
</tbody>
</table>

### Monitoring Service

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event History Monitor: Error</td>
<td>An error was encountered by the Event History Monitor, the process responsible for historical data collection from various sources.</td>
</tr>
<tr>
<td>Job Monitor: Error</td>
<td>An error was encountered by the Job Monitor, the process responsible for collecting data for running SQL Server Agent Jobs.</td>
</tr>
<tr>
<td>Message Processor: Message Action Throttled</td>
<td>The Message Processor reached its max messages per interval threshold, and at least one message was throttled. Subsequent messages will be sent to the Actions Log. See SentryOne Monitoring Service-&gt;Settings.</td>
</tr>
<tr>
<td>Performance Monitor: Error</td>
<td>An error was encountered by the Performance Monitor, the process responsible for collecting performance data.</td>
</tr>
<tr>
<td>Reporting Services Monitor: Error</td>
<td>An error was encountered by the Reporting Services Monitor, the process responsible for collecting data for SSRS reports.</td>
</tr>
<tr>
<td>Monitoring Service: Action Failed</td>
<td>The SentryOne Monitoring Service attempted to execute an action, but the action failed. For example, the Send Email action can fail if the SMTP server is unavailable.</td>
</tr>
<tr>
<td>Monitoring Service: SentryOne Database Offline</td>
<td>The SentryOne database is offline or otherwise inaccessible.</td>
</tr>
<tr>
<td>Monitoring Service: SentryOne Database Online</td>
<td>The SentryOne database is online and accessible.</td>
</tr>
</tbody>
</table>
Monitoring Service: Performance Collection Failed
The Performance Monitor encountered an error while collecting data for a specific performance counter.

Monitoring Service: Repository Error
An error occurred while writing data to the SentryOne database.

Monitoring Service: Synchronization Error
The SentryOne Monitoring Service encountered a critical error while synchronizing settings from the SentryOne database with its configuration management thread.

Windows Task Scheduler Monitor: Error
An error was encountered by the Task Scheduler Monitor, the process responsible for collecting data for Windows Tasks.

**SQL SERVER INSTANCE**

Remember to make use of Condition Settings when configuring alerts for Conditions. Condition filters are available which can help fine tune exactly when a condition is considered met.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Agent: Offline</td>
<td>SQL Server Agent is offline or otherwise inaccessible.</td>
</tr>
<tr>
<td>SQL Server Agent: Online</td>
<td>SQL Server Agent is online and accessible.</td>
</tr>
<tr>
<td>SQL Availability Replica Healthy</td>
<td>The SQL Availability Replica is healthy.</td>
</tr>
<tr>
<td>SQL Availability Replica Unhealthy</td>
<td>The SQL Availability Replica is unhealthy.</td>
</tr>
<tr>
<td>SQL Server Cluster Failover</td>
<td>The SQL Server cluster failed over.</td>
</tr>
<tr>
<td>SQL Server Instance: Offline</td>
<td>SQL Server is offline or otherwise inaccessible.</td>
</tr>
<tr>
<td>SQL Server Instance: Online</td>
<td>SQL Server is online and accessible.</td>
</tr>
</tbody>
</table>

**WINDOWS INSTANCE**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td>Windows Instance is offline or otherwise inaccessible.</td>
</tr>
<tr>
<td>Online</td>
<td>Windows Instance is online and accessible.</td>
</tr>
</tbody>
</table>

7.1.4 Audit Conditions
Audit Conditions are applicable to various activities carried out both inside and outside of the SentryOne Client. Audit Conditions provide a way for you to be notified when various Settings are changed, or monitored objects are modified from within the SentryOne Client.

💡 The Log to Database Action is configured by default for all Audit Conditions. This gives you an audit trail concerning changes to your SentryOne environment. If you would like to be notified about an Audit Conditions happening in your environment you could configure the Send Email Action.

CONFIGURE ACTIONS FOR AUDIT CONDITIONS

Configure Actions for Audit Conditions

You can configure additional Actions in response to Audit Conditions being met at these levels: Shared Groups (Global), Site, Computer Group, Computer, and Connection.

To do so, select the node appropriate to the level you would like to configure the action for in the Navigator Pane and then the Audit Actions tab in the Conditions pane. If you do not see the Conditions pane use the View menu → Audit Actions.

Next you will want to click the Add button found in the Conditions pane. This will open the Select Action window. Expand the applicable Object and Condition. Use the check box(s) to select which Actions should be taken in response to this Condition being met. Click the OK button.

For more information about Actions that can be taken when a Condition is met see the Actions topic.

For a complete listing of all Audit conditions, see below.

Audit Conditions

Custom Event View

Event Chain

Global

Performance Counter

CUSTOM EVENT VIEW

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Event View Created</td>
<td>A Custom Event View was created.</td>
</tr>
<tr>
<td>Custom Event View Deleted</td>
<td>A Custom Event View was deleted.</td>
</tr>
</tbody>
</table>

EVENT CHAIN

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Chain Modified</td>
<td>An Event Chain was modified.</td>
</tr>
</tbody>
</table>

GLOBAL

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions Changed</td>
<td>A configured Action was changed for an object.</td>
</tr>
<tr>
<td>Contacts Changed</td>
<td>A Contact was changed.</td>
</tr>
</tbody>
</table>
Event Object Created

An Event Object (job, task, etc.) was created.

Event Object Deleted

An Event Object (job, task, etc.) was deleted.

Event Object Properties Modified

An Event Object's (job, task, etc.) properties were modified.

Event Object Rescheduled

An Event Object (job, task, etc.) was rescheduled. For some object types this condition is only detected when rescheduling in the SentryOne Client via drag-and-drop.

Notes Changed

A note was changed.

Response Ruleset Created

A Response Ruleset was created.

Response Ruleset Deleted

A Response Ruleset was deleted.

Response Ruleset Modified

A Response Ruleset was modified.

Settings Changed

Object or application settings were changed.

Watched Flag Changed

The watched flag was changed for an object.

**PERFORMANCE COUNTER**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Counter Added</td>
<td>A Performance Counter was added to an event object or event view.</td>
</tr>
<tr>
<td>Performance Counter Deleted</td>
<td>A Performance Counter was deleted from an event object or event view.</td>
</tr>
<tr>
<td>Performance Monitored Flag Changed</td>
<td>The Performance Monitored flag was changed for an event object or event view.</td>
</tr>
</tbody>
</table>

### 7.1.5 Advisory Conditions

**Advisory Conditions** are a Performance Analysis feature that allows for customized alerting based on performance counter metrics, query results, and duration of events. **Advisory Conditions** can be applied globally or at specific Targets and Instances.

⚠️ **NOTE:** A license for Performance Analysis is required for Advisory Conditions to be evaluated.

**DOWNLOADING THE ADVISORY CONDITIONS PACK**

When opening the **Conditions List** at the Global level, you will be prompted to download the latest Advisory Conditions Pack. This pack contains predefined Advisory Conditions that will help illustrate the functionality that the Advisory Conditions feature has to offer. If you decide to not
download the Advisory Conditions Pack at this point, you can always access it through the Download Latest Advisory Conditions Pack option in the Tools menu. Also, if you chose to disable the automatic checking for Advisory Conditions Packs, checking can be re-enabled in the User Preferences in the Updates section.

After downloading the Advisory Conditions Pack, you will be prompted to apply actions. Selecting yes will open the Actions Selector. Here you can apply actions to your new Advisory Conditions.

**NOTE:** Scheduled evaluations will not occur until an action is assigned to the Advisory Condition.

### BUILDING AN ADVISORY CONDITION

To begin creating an Advisory Condition, you can either click the Create Advisory Condition button in the Conditions List or you can right click on the All Targets (Global) node, a group node, a Target node, or an Instance node in the Navigator Pane and select the Add Advisory Condition option. Please note that the owner object of the Advisory Condition is dependent on where the Advisory Condition is defined. The following sections will explain all of the options available when creating an Advisory Condition.

### WHERE TO DEFINE AN ADVISORY CONDITION

In the Navigator Pane, there are Advisory Condition folders located at various levels. The chart below outlines the various levels and what types of conditions can be configured at those levels.

<table>
<thead>
<tr>
<th>Location in Hierarchy</th>
<th>Advisory Condition Types available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root/Global/Site/Target</td>
<td>Windows</td>
</tr>
<tr>
<td></td>
<td>VMware Host</td>
</tr>
<tr>
<td></td>
<td>Windows/SQL Server</td>
</tr>
<tr>
<td></td>
<td>Azure SQL Database</td>
</tr>
<tr>
<td></td>
<td>Windows/SSAS</td>
</tr>
<tr>
<td></td>
<td>SentryOne</td>
</tr>
<tr>
<td>SQL Server Instance</td>
<td>Windows/SQL Server</td>
</tr>
<tr>
<td>SSAS Instance</td>
<td>Windows/SSAS</td>
</tr>
</tbody>
</table>

While SQL Server and SSAS Advisory Conditions can be defined higher in the hierarchy, defining them at the Instance provides the added benefit of being context aware allowing you to utilize performance counter instances specific to that Instance and execute SQL queries within the context of that Instance.

### ADVISORY CONDITION TYPES

The first option presented when creating an Advisory Condition is the type. The type determines where the condition will be evaluated and what metrics are available.

- **Windows** - This condition will be evaluated at the Target level and has access to Windows performance metrics.
- **SQL Server** - This condition will be evaluated at the SQL Server Instance level and has access to Windows and SQL Server performance metrics.
- **Azure SQL Database** - This condition will be evaluated at the Azure SQL Database level and has access to Azure SQL Database performance metrics.
- **VMware Host** - This condition will be evaluated at the VMware Host level and has access to vCenter performance metrics associated with a VMware Host.

- **Analysis Services** - This condition will be evaluated at the SSAS Instance level and has access to Windows and SSAS performance metrics.

- **SentryOne** - This condition type will be evaluated against the SentryOne repository and is used to query the repository for information.

**ADVISORY CONDITION OPTIONS**

**Default Evaluation Frequency** – Defines how often the condition is evaluated. The evaluation frequency can be adjusted individually throughout the hierarchy.

**NOTE:** Scheduled evaluations will not occur until an action is assigned to the Advisory Condition.

**Trigger Threshold** – Defines how long the condition must be true before the associated action is taken.

**Severity** – Defines the setting of severity, which can be used as a sorting parameter in the Events Log. Advisory Conditions with a severity of High or Critical will also cause the Advisory Conditions folder in the Navigator pane to turn red when true.

**Evaluation Timeout** – Defines the limit on how long each evaluation is allowed to run before a timeout occurs.

**Maximum Instance Count** - Defines the number of results returned when the Any instance option is used.

**Color** – Defines the color of the duration line that appears on the Performance Analysis Dashboard.

**SUPPORTED VERSIONS**

When creating an Advisory Condition in the higher levels of the hierarchy, you may want to restrict certain conditions to run only on designated versions of Windows and/or SQL Server. For example, if you wanted to restrict the condition to only evaluate on servers with Windows Server 2008 R2 and newer, you would put 6.1 in the Minimum field.

**DEFINING AN ADVISORY CONDITION**

Select a logical operator.

- **And** - All conditions must be true for an action to occur
- **Or** - One or more conditions must be true for an action to occur
- **Not And** - One or more conditions must be false for an action to occur
- **Not Or** - All conditions must be false for an action to occur

Click the Plus icon and select a comparison type or an existing Advisory Condition.

💡 If an Advisory Condition is altered, it is altered in all other Advisory Conditions that are using it as an existing condition.

The following table shows what value retriever types are available for the various comparison types. Gray cells indicate that the value retriever type is unavailable for that comparison type.
### VALUE RETRIEVER TYPES

**Performance Counter** – Performance counters are available based on the type of condition that is being created. Conditions being built at the Target and Instance level are context aware and will have access to specific counter instances. When defined at the Target level or lower, you can also use baselines that are associated with performance counters.

<table>
<thead>
<tr>
<th>Type</th>
<th>Numeric Comparison</th>
<th>String Comparison</th>
<th>DateTime Comparison</th>
<th>Additional AC Type Restrictions</th>
<th>Expression Left Side</th>
<th>Expression Right Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Counter</td>
<td></td>
<td></td>
<td></td>
<td>Performance Counter instances are only available at the appropriate instance context.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Counter Baseline</td>
<td></td>
<td></td>
<td></td>
<td>User defined Baselines must exist on the Instance.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Server Query</td>
<td></td>
<td>Windows/SQL Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repository Query</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Azure SQL Database Query</td>
<td></td>
<td>Azure SQL Database Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware Host</td>
<td></td>
<td>VMware Host only</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMI Query</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explicit Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expression</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existing Condition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** SentryOne provides virtual performance counters for use in Advisory Conditions. These virtual counters are gathered using stored procedures and may cause more overhead than counters collected through PerfLib. Virtual performance counters can be identified by the lack of a colon. For example, SQL Server Data File Size is a virtual counter.

**Performance Counter (User)** - The Performance Counter (User) option allows for the selection of any performance counter. After selecting this option, click the Connect button, enter a server name, then click Connect again. After connecting to the server, select the Category field to view a list of all of the performance counter categories on that server. After selecting a category, select the counter that is to be evaluated.

**NOTE:** Connecting to a server is only necessary to provide a list of performance counter categories and counters. During evaluation, the counter will be evaluated in the context of each monitored Instance.

**Performance Counter Baseline** - Performance Counter Baselines are available when the Advisory Condition is created at the Target or Instance level. A user defined baseline must exist at the Instance level to be used.

**SQL Server Query** – Enter the database that the query will be run against in the first field. If the condition is being defined at the Instance level, there will be a button next to the field that provides a list of databases available for that Instance. The next field is for the query. The query can be entered manually or can be selected from a list of queries using the Query Editor. In the event that a query returns multiple results, the first result is used for the evaluation.

**Azure SQL Database Query** – Enter the database that the query will be run against in the first
field. The next field is for the query. The query can be entered manually or can be selected from a list of queries using the Query Editor. In the event that a query returns multiple results, the first result is used for the evaluation.

**Repository Query** - This option allows you to perform a query against the SentryOne repository. When using a Repository Query, you can use the variables @ComputerID and @ConnectionID in your query to dynamically adjust depending on the Computer or Connection that the Advisory Condition is evaluating against.

**WMI Query** – This option allows you to query WMI using WQL. For additional information, please see the following Microsoft article, [Querying with WQL](#).

**Explicit Value** – This allows for an explicitly defined value.

**Last Value** – This is a right-side specific option that returns the previous evaluations value of the left side of the argument.

**Expression** – Using the Expression option allows for basic mathematical operation to be applied to other values. Operations include addition, subtraction, multiplication, and division. These mathematical operations can be performed against any of the other numeric value options.

**Duration** – This option returns the amount of time elapsed to acquire the specified value. Various time units are available, ranging from milliseconds to days.

**NOTE:** When using any query retriever type in conjunction with the “Any” qualifier to compare multiple results, the query is required to return a key/value pair in order to identify which value met the defined condition. This also ensures that conditions using queries on both sides of the condition, or comparing Last Value, will automatically match values with the same key.

### QUERY EDITOR

When using a Query retriever type (such as the SQL Server or Repository query), you can click the button to the right of the query input box to launch the Query Editor.

The Query Editor is where queries can be edited and saved for future use in Advisory Conditions.

**New** – Open the New Query window.

**Test** – Opens a Instance dialog window. Specify the server name and the database name that the query is to be executed against. The resulting value will be displayed in the field to the right of the test button.

**Delete** – Delete the selected query.

**NOTE:** Only queries that have been tested and have a Value Data Type can be selected for use from the Query Editor.

### CONDITIONS LIST

The Conditions List can be accessed from various locations within the hierarchy of the Navigator Pane. The number in parenthesis beside the Advisory Conditions folder indicates how many Advisory Conditions are defined at that level within the hierarchy. An exception to this is Advisory Conditions folder located beneath the Contacts node in the Navigator Pane. The number here indicates the total number of Advisory Conditions defined across the environment.

What is seen in the Conditions List depends on where the list is opened within the hierarchy. Wherever the list is opened, the list will contain Advisory Conditions defined at that level, its parents’ levels and children’s levels.
**CONDITIONS PANE**

The Conditions Pane provides a list of all available conditions. This pane also displays the Owner Object, creation date, and last modified date of the condition as well as what user modified it. You can disable an Advisory Condition by unchecking the checkbox in the Enabled column. You can also add Tags to assist with sorting and organizing the conditions.

The Create Advisory Condition button allows for the selection of Windows, Windows/SQL Server, or Windows/SSAS. After selecting the condition type, the Condition Editor will be opened.

The Edit Advisory Condition button opens the Condition Editor for the condition that is currently selected in the Conditions Pane.

The Show Events Log button takes you to the Events Log filtered by the Advisory Condition that is selected in the Conditions Pane.

**Context Menu**

- **Edit** – Allows for editing of the selected Advisory Condition
- **Show Events Log** - Opens the Events Log filtered to the selected Advisory Condition
- **Snooze**
  - **This Condition/All Objects** – Disables actions and alerting on the selected Advisory Condition for the selected period of time
  - **All Conditions/All Objects** – Disables actions and alerting on all Advisory Conditions for the selected period of time
- **Unsnooze**
  - **This Condition/All Objects** – Re-enables actions and alerting on the Advisory Condition
  - **All Conditions/All Objects** – Re-enables actions and alerting for all Advisory Conditions
- **Clone** – Creates a copy of the selected Advisory Condition with a new name with the Owner Object being in the context of the Conditions List
- **Export** – Exports the selected Advisory Condition to a .condition file
- **Import** – Imports an Advisory Condition from a .condition file
- **Disable** – Disables the selected Advisory Condition
- **Delete** – Deletes the selected Advisory Condition

**EVALUATION STATUS PANE**

The Evaluation Status Pane displays the last evaluation for the selected condition. This pane shows the Object that was evaluated, the result of the evaluation, the duration and time of the evaluation, and information pertaining to any errors that may have occurred. The bottom of the pane provides a count of the objects that were evaluated. Selecting an evaluation in this pane will display the results of that evaluation in the Evaluation Results Pane below.

The Evaluate Selected button will perform an evaluation of the Advisory Condition against the selected Object in the Evaluation Status Pane. The **With Logging** option will generate an entry in the Events Log if the condition does not have the Log to Alerting Channels action assigned to it.

The Evaluate All button performs an evaluation of the Advisory Condition against all legal targets. The **With Logging** option will generate an entry in the Events Log if the condition does not have
the Log to Alerting Channels action assigned to it.

**Context Menu**

- **Snooze -> This Condition/This Object** – Disables actions and alerting for the selected Advisory Condition on the selected object for the selected amount of time
- **Jump To Performance Analysis Dashboard** – Opens Performance Analysis for the selected object

**EVALUATION RESULTS PANES**

The Evaluation Results Pane, located at the bottom, shows the evaluation step results of an Advisory Condition along with any values retrieved through performance counters or queries. Mousing over any of the values will provide a tooltip with additional information like the duration of retrieving that value, the non-rounded value, error messages, and performance counter instance names.

**EVENTS LOG**

The Events Log displays all of the Advisory Condition events that are active or have completed. Selecting an entry in the grid view will display the starting state of the event. In the case of completed Advisory Conditions, clicking on the End Time column will display the result state that closed the active event. By clicking the expansion node in the Notes column, you can add notes to the event which can also be included in future notifications of that condition.

You also have the ability to assign users to events. To do this, click into the Assigned User column and select a user from the dropdown list. Alternatively, you can automatically assign a user to all events of a particular Advisory Condition by selecting an Advisory Condition in the Conditions pane that has the **Send to Alerting Channels** action assigned to it and selecting the user in the Action Settings tab. Assigning users to Advisory Condition events allows for users to quickly sort events and investigate any events that they have been assigned.

**Context Menu**

- **Jump to ->**
  - **Performance Analysis Dashboard** – Jumps to the time frame of the Advisory Condition event
  - **Navigator** – Jumps to the Target or Instance where the Advisory Condition event was evaluated
- **Edit Advisory Condition** – Opens the Advisory Condition Editor for the selected Advisory Condition
- **Disable/Enable Advisory Condition** – Disables or Enables the selected Advisory Condition
- **Snooze ->**
  - **This Condition/This Object** – Disables actions and alerting on the selected Advisory Condition on the selected object for the selected period of time
  - **This Condition/All Objects** – Disables actions and alerting on the selected Advisory Condition on all objects for the selected period of time
The Alerting and Response system in SentryOne uses the concept of **Conditions** and **Actions**. **Actions** can be defined in response to certain **Conditions** being met within your environment. You can choose from a variety of **Actions**, depending on which **Condition** is being responded to.

All **Actions** work on the principle of inheritance. This means that if you configure an **Action** in response to a **Condition** being met at the Global level (All Targets), it will be automatically passed down to all applicable objects below it. This allows you to define Global **Actions** for the most common issues across your environment once, and have those passed down to every monitored server automatically. You can further refine **Actions** at each level as needed. For a visual representation of how inheritance works within SentryOne see the Alerting and Response System Hierarchy diagram.

Each **Action** that you configure in your environment will have an associated behavior. The behavior controls how the **Action** will be carried out relative to any inherited **Actions**. There are three Action behaviors available: **Override Inherited Actions**, **Combine with Inherited Actions**, and **Disabled**. For a complete explanation and example usage scenario for each behavior see the **Action Behavior** section below.

You can fine tune **Actions** for a specific group of objects or even an individual object, giving you effective control over how SentryOne works throughout your environment. SentryOne comes with a number of Global **Actions** predefined to get you up and running quickly. These **Actions** can be changed, as needed to fit the specific needs of your environment.

**Response Rulesets** control how often **Actions** are taken in response to **Conditions** being met. They do this by assigning additional criteria on top of the Condition itself, that must be met before an Action will take place. To control the time frame of when an **Action** will be processed you may also apply a **Window** directly to any configured Action.

### Jump to:

- **Actions**
- **Kill Task**
- **Log to Event Log**
- **Log to Disk**
- **Send Email**
- **Execute SQL**
- **Send SNMP Trap**
- **Log To Database**
- **Execute Process**
- **Send Page**
- **Run QuickTrace**
- **Execute Job**

### HOW TO CONFIGURE ACTIONS

If you are upgrading from a previous version of SentryOne you will notice that the Actions and
Settings pane has been overhauled.

The Conditions displayed in the Conditions pane will change depending on which node or object you have selected in the Navigator pane. If you do not see the Conditions pane once you have selected a node in the Navigator pane, use the View Menu (View → Conditions).

If you select the All Targets node you will see globally applied Conditions in the Conditions pane. When you select any applicable object level below the All Targets node, you will see two specific sets of applied Conditions in the Conditions pane.

The top section of the Conditions pane contains the Inherited Section, which shows you any applied Conditions that are being passed down to the current object level. The Inherited Section also contains an Object column, which shows you the object level the Condition is being inherited from. When an Inherited Condition is overridden or disabled, it will still show up in the Inherited Section, but its text will be grayed-out, and its status will say Overridden.

Directly beneath the Inherited Section, is the Explicit Section. The Explicit Section contains applied Conditions that have been set at the current object level. The Explicit Section also displays a Behavior column. Each Action that you set up in your environment will have an associated behavior. This behavior controls how the Action will be carried out relative to any inherited Actions.

To add a new Action, select the desired node in the Navigator pane. Next you will want to click the Add button found in the Conditions Pane. This will open the Action Selector window. Expand the applicable object and Condition. Use the check box(s) to select which Actions should be taken in response to this Condition being met. Click the OK button.

You may also choose to quickly Disable, Override, or Combine any inherited Actions. To do so, simply select a Condition in the Inherited Section of the Conditions pane and choose the desired command (Disable button, Override button, or Combine button).

**ACTION BEHAVIORS**

Each action that you setup in your environment will have an associated behavior. This behavior controls how the action will be carried out relative to any inherited actions.

The user can change the Behavior of any Action to one of the following:
<table>
<thead>
<tr>
<th>Behavior</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Override Inherited Actions</strong></td>
<td>Actions defined with this behavior will override all actions that are being inherited from a higher level. This behavior can be thought of as a special set of instructions which are followed instead of the passed down (inherited) instructions.</td>
</tr>
<tr>
<td></td>
<td>When a Condition occurs which triggers an action defined with the Override Inherited Actions behavior, inherited actions will not be executed, but the action which is defined at that specific level will be executed.</td>
</tr>
<tr>
<td></td>
<td>The Override Inherited Actions behavior is the default behavior assigned for any newly defined action.</td>
</tr>
<tr>
<td></td>
<td>→ For an example scenario see the Override Inherited Actions Example.</td>
</tr>
<tr>
<td><strong>Combine with Inherited Actions</strong></td>
<td>Actions defined with this behavior will be combined with all inherited actions. This behavior can be thought of as a set of instructions which are followed in addition to the passed down (inherited) instructions.</td>
</tr>
<tr>
<td></td>
<td>When a Condition occurs which triggers an Action defined with the Combine with Inherited Actions behavior, all inherited actions will be executed, and all actions that are defined at that specific level will also be executed.</td>
</tr>
<tr>
<td></td>
<td>→ For an example scenario see the Combine with Inherited Actions Example.</td>
</tr>
<tr>
<td><strong>Disabled</strong></td>
<td>When an action is defined with the Disabled behavior it will disable any action of the same action type, which is being inherited for a certain Condition. This behavior can be thought of as a special set of instructions which simply disallow the passed down (inherited) set of instructions.</td>
</tr>
<tr>
<td></td>
<td>When a Condition occurs which triggers an Action defined with the Disabled behavior, inherited Actions will not be executed.</td>
</tr>
<tr>
<td></td>
<td>→ For an example scenario see the Disabled Example.</td>
</tr>
</tbody>
</table>

**MULTIPLE ACTIONS PER CONDITION**

One of the new improvements made to the Alerting and Response System is the ability to add multiple versions of the same action type per condition. You can use this ability, in combination with Response Rulesets to implement escalation procedures when certain Conditions take place.

⚠️ **NOTE:** Configured actions of the same action type, defined at the same object level, will have their behavior types bound together. This is because inherited actions can not
simultaneously be in more than one state. (disabled, combined, and overridden)

**ACTIONS**

**KILL TASK ACTION**

The **Kill Task Action** will "Kill", or cancel the current job. This action is typically associated with Runtime Maximum, Performance Threshold Maximum, Block, or Conflict conditions, in order to prevent resource contention situations. It is not applicable to the Started condition, or any conditions implying event instance completion.

**LOG TO WINDOWS EVENT LOG ACTION**

The log to **Event Log Action** sends data relevant to the condition to the Application Event Log on the SentryOne Monitoring Service computer.

**LOG TO DISK ACTION**

The **Log to Disk Action** sends data relevant to the condition to the specified text file on the SentryOne Monitoring Service computer’s file system.

**SEND EMAIL ACTION**

The **Send Email Action** sends an email alert to the Users or Groups specified in the Action Settings. For more information about Users and Groups see the Contact Management topic.

Certain Alert emails in version 7.0 now include a [View in SentryOne Client] section. You will notice that the Event View section contains a hyperlink which begins with the SentryOne URL protocol. Selecting this link will open the Event View in the SentryOne Client for the specific condition which triggered the alert.

⚠️ To stop receiving the Outlook Security Notice when you select an Alert email link you will need to enable trust for the SentryOne protocol. Please see the Trusting the SentryOne hyperlink in Outlook KB article.

**Settings**

The Actions Settings tab allows the following configurations.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Targets</td>
<td>Specify which Users or Groups will receive the email alert.</td>
</tr>
<tr>
<td>Importance</td>
<td>Specify the importance level of the email alert.</td>
</tr>
<tr>
<td>From Address</td>
<td>Specify the from email address of the alert. To use the default email address leave this blank. For more information see the SMTP configuration section of the SentryOne Monitoring Service Settings topic.</td>
</tr>
</tbody>
</table>

**EXECUTE SQL ACTION**

The **Execute SQL Action** executes the T-SQL statement(s) on the specified SQL Server.
**Settings**

Use the Server list to select a server, and enter the desired T-SQL in the command text box. Selecting the (Target) option will execute the T-SQL statement against the server that triggered the condition.

**NOTE:** In an environment with multiple monitoring services, it is important to note that the service that detects the condition is the service that will fire the action. If that service does not have connectivity to the targeted server, the action will fail.

---

**SEND SNMP TRAP ACTION**

The **Send SNMP Trap Action** sends an Simple Network Management Protocol (SNMP) trap notification. For more about configuring SNMP see the SentryOne Monitoring Service Settings topic.

---

**LOG TO DATABASE ACTION**

The **Log To Database Action** will log data relevant to the condition to the Actions log.

---

**EXECUTE PROCESS ACTION**

The **Execute Process Action** will execute the defined command text on the specified SQL Server.

**Settings**

Within the Execute Process Action Settings tab you will see the Execute Process section. Use the Server list to select a server. Selecting the (Target) option from the server list will execute the process against the server that triggered the condition. Use the Command Text box to enter the desired literal command.

**NOTE:** You must enable xp_cmdshell to use this action. You may also pass certain system parameters, as well as user defined parameters through output text.

---

**SEND PAGE ACTION**

The **Send Page Action** sends an alert to the Pager address for the specified Users or Groups. For more information about Users and Groups see the Contacts Management topic.

**Settings**

The Actions Settings tab for the Send Page Action allows the following configurations.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Targets</td>
<td>Specify which Users or Groups will receive the pager alert.</td>
</tr>
<tr>
<td>Importance</td>
<td>Specify the importance level of the pager alert.</td>
</tr>
<tr>
<td>From Address</td>
<td>Specify the From address of the alert. To use the default email address leave this blank. For more information see the SMTP configuration section of the SentryOne Monitoring Service Settings topic.</td>
</tr>
</tbody>
</table>

---

**RUN QUICKTRACE™ ACTION**

The **Run QuickTrace Action** executes a Quick Trace™ against a specified target server. A Quick
Trace is a comprehensive snapshot of activity created by combining process-level data and trace events collected during a brief sample period. A Quick Trace is not filtered, so it collects all events on the specified SQL Server.

💡 The QuickTrace Action is now available for a wide range of conditions.

To avoid impacting the performance of the target SQL Server, Quick Trace is both time and size limited. Only one simultaneous Quick Trace can be run against any target server. Quick Traces are also throttled to control the time between successive runs. The current allowed frequency is one minute. This means that once a Quick Trace is completed, another Quick Trace cannot begin on the same specified server until one minute has elapsed.

Additionally, in order to avoid impacting server performance on very busy systems, there are certain cases where SentryOne will further restrict Quick Trace functionality. If one of these cases is detected the QuickTrace Action will not fire. Please see Quick Trace Restrictions for more information about these cases.

**Settings**
The Actions Settings tab for the QuickTrace Action allows the following configurations.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run QuickTrace Against</td>
<td>Specify the Target server on which the Quick Trace will run.</td>
</tr>
<tr>
<td>Run QuickTrace for</td>
<td>Specify the length of time which the Quick Trace will run against the target server.</td>
</tr>
<tr>
<td>Collect Statement Events</td>
<td>Specifies whether or not to collect Statement Level Events. Enabling the Collect Statement Events option will dramatically increase the amount trace data collected.</td>
</tr>
<tr>
<td>Limit Trace Data to</td>
<td>Specify the maximum number of rows which the Quick Trace will collect.</td>
</tr>
</tbody>
</table>

**EXECUTE JOB ACTION**
The Execute Job Action executes a SQL Agent job on the current server, or any other watched server in your enterprise. This action effectively enables you to create simple “one-to-one” job chains by associating it with Completed, Success or Failure conditions. Also see Event Chains for more advanced chaining features.

**Settings**
Within the Action Settings tab you will see the Jobs To Execute section. Use the server list to select a server. Use the job list to specify the SQL Agent Job to run. Once a target job has been specified for this action, it will show up in the Event Calendar Calendar View with a grid pattern background as an indication of its conditional execution.

**EXECUTE POWERSHELL ACTION**
The Execute PowerShell Action executes the entered PowerShell script on the monitoring computer, the target computer, or any valid Windows target.
**NOTE:** You may also use system parameters in the PowerShell script, with a format of `<%ParameterName%>` (e.g, `$ServerName = '<%ServerName%>'`).

**Settings**

Within the Action Settings tab you will see the **Server** section. Use the server list to select where the script will be executed: on the monitoring server, on the target server, or on a specific watched server.

Use the account list to select whether the defined PowerShell Execution Account, a Domain Account, or a Local Account will be used to run the script. Note that a valid user name and password must be provided in the Action Settings if the Domain Account or Local Account options are selected.

Enter or load (using the Load from File button beneath the text area) PowerShell script text into the **PowerShell Script Text** section; the Test PoSh Script button can be used to test the entered script on a specified target.

**SEND TO ALERTING CHANNELS**

The **Send to Alerting Channels** action allows open Advisory Condition events to be displayed in various place throughout the client.

Selecting the Performance Analysis Dashboard option will show the open Advisory Condition events on the Performance Analysis Dashboard if the Advisory Condition involves a performance counter that is associated with one of the charts on the Dashboard. A glyph will also appear in the top left corner of the chart and provide you with additional information pertaining to the Advisory Condition event.

Additionally, you can select a user to be automatically assigned to an Advisory Condition event by selecting a user from the dropdown list that appears in the Action Settings tab of the Conditions pane.

The blog post [Advisory Conditions : Available Actions](#) contains additional information and examples on this topic.

---

### 7.2.1 Action Behaviors

**OVERRIDE INHERITED ACTIONS EXAMPLE**

Actions defined with the **Override Inherited Actions** behavior will override all actions that are being inherited from a higher level. This behavior can be thought of as a special set of instructions which are followed instead of the passed down(inherited) instructions.

For this example assume the following:

There is a **Send Email Action** configured at the **Global level** which has a selected target of **DBA1** for the **SQL Server Agent Job: Failure condition**.

With this Action configured as above, anytime a SQL Server Agent Job **fails across the entire monitored Enterprise** the following will occur:

1. The **SQL Server Agent Job: Failure** condition is met.
2. In response to the Condition being met, the Send Email Action will execute, sending a notification email to DBA1.

Now suppose there is a Development Server (DEVServer) in the environment, which is not managed by DBA1. For this particular server when an Agent Job fails, instead of the user DBA1 being alerted, you would like notification emails sent to the developer or user DEV1.

To accomplish this you would want to add a Send Email Action with a selected target of DEV1, for the SQL Server Agent Job: Failure condition at the Instance level with a behavior of Override Inherited Actions.

With this Action configured as above, anytime a SQL Server Agent Job fails on the Development Server the following will occur:

1. The SQL Server Agent Job: Failure condition is met.
2. In response to the condition being met, the Send Email Action will execute, sending a notification email to DEV1

Only the Development Server would be impacted by this change. The rest of the monitored Enterprise would still be subject to the Globally configured Send Email Action which has a selected target of DBA1.

**COMBINE WITH INHERITED ACTIONS EXAMPLE**

Actions defined with the Combine with Inherited Actions behavior will be combined with all inherited actions. This behavior can be thought of as a set of instructions which are followed in addition to the passed down (inherited) instructions.

For this example assume the following:

There is a Send Email Action configured at the Global level which has a selected target of DBA1 for the SQL Server: Offline condition.

With this Action configured as above, anytime a SQL Server is detected to be offline across the entire monitored Enterprise, the following will occur:

1. The SQL Server: Offline Condition is met.
2. In response to the Condition being met, the Send Email Action will execute, sending a notification email to DBA1.

Now suppose there is a SQL Server (ASPServer) in the environment that has an internet facing workload. When the SQL Server: Offline Condition is met you would still like to notify DBA1, but you would also like to notify the Web Developer, WEBDEV1.

To accomplish this you would want to add a Send Email Action, with a selected target of WEBDEV1, for the SQL Server: Offline condition at the Instance level with a behavior of Combine with Inherited Actions.

With this Action configured as above, anytime that the ASPServer is detected to be offline, the following will occur:

1. The SQL Server: Offline Condition is met.
2. In response to the condition being met, the Send Email Action will execute, sending a notification email to DBA1 and WEBDEV1.
Only the ASPServer would be impacted by this change. The rest of the monitored Enterprise would still be subject to the Globally configured Send Email Action for the SQL Server: Offline Condition, which has a selected target of DBA1.

**DISABLED BEHAVIOR EXAMPLE**

When an Action is configured for a Condition with the Disabled Action behavior, it will disable any action of the same action type, which is being inherited. This behavior can be thought of as a special set of instructions which simply disallow the passed down (inherited) set of instructions.

For this example assume the following:

On TestServer1 there is a Send Email Action configured which has a selected target of DBA1 for the SQL Server Agent Job: Runtime Threshold Max condition. This action is configured at the Instance level.

With this Action configured as above, anytime any SQL Agent Job meets the Runtime Threshold Max condition on TestServer1 the following will occur:

1. The SQL Server Agent Job: Runtime Threshold Max Condition is met.
2. In response to the Condition being met, the Send Email Action will execute, sending a notification email to DBA1.

Now suppose there is a Job, LongRunningJob, that is known to be long running on TestServer1. DBA1 has no need to be alerted about it and would like to disable runtime notifications about this job.

To accomplish this you would want to add a Send Email Action for the SQL Server Agent Job: Runtime Threshold Max condition at the object level (the LongRunningJob) with a behavior of Disabled.

With this Action configured as above when the LongRunningJob runs long, the following will occur:

1. The SQL Server Agent Job: Runtime Threshold Max Condition is met.
2. In response to the Condition being met, the inherited Send Email Action will not be executed.

Only the individual SQL Agent Job LongRunningJob will be impacted by this change. The rest of the SQL Agent Jobs on TestServer1 will still be subject to the Instance level Send Email Action configured for the SQL Server Agent Job: Runtime Threshold Max Condition.

7.2.2 Message Editing

This feature allows you to fully customize the messages that are generated by SentryOne. This gives you the ability to add or remove information, change how the information is formatted, and change the order in which the information is displayed. To access the Message Editor, select the condition that you'd like to change in the Conditions Pane, navigate to the Message Tab in the bottom of the pane, then click the Edit button.

**HOW DOES CUSTOMIZATION WORK?**
First, it is important to understand the format of the XML that describes the format of the messages, the elements that can be added to the XML and how these elements are related to one another.

At the root level (the "message" element), each message format XML document (called a "template") can use the "imports" attribute to inherit from another template. That template can in turn inherit from another template and so on creating a chain of parent/child relationships. These chains are used extensively throughout the template files to reduce the repetition of reused elements. The middle pane of Message Editor coalesces these chains into the single, complete XML document that will be used to generate the final message. Sometimes the finale template, sometimes referred to as a leaf template, has very little information because almost all of the XML is in the inherited templates. In these cases, it can be very useful to look at the Complete XML to figure out what is in the message and how to change it.

Inside the root message element, there are three parts to a template: the format element, the tokenSets element and the body element. The format element contains definitions for section, item and token styles. The tokenSet element contains a list of all of the tokenSets associated with the template. The body element contains all of the sections, items and tokens that describe the message format itself. All of these elements are combined with their imported parent elements during the import process so that inheritance can be used. The format element is a list of style definitions (discussed below) that can be referred to in the body of the template. Leaf templates often don’t contain a format element because they rely on a group of standard, inherited styles. However, you can add a format element if one does not exist or add to an existing one to define new styles. Only one format element is allowed per template.

The tokenSets element is a list of all of the tokenSets associated with a template. A tokenSet is a logical grouping of tokens that exists in SentryOne. You cannot add tokenSets to messages because the data for the tokens in the tokenSet needs to be created and assigned in the program. A list of all of the tokens in the tokenSets is available in the Message Editor. Only one tokenSets element is allowed per template.

The body element describes all of the "things" that a message is made of. These things can be categorized as items (and things that inherit from items) and sections, which are essentially groups of items. Only one body element is allowed per template.

ITEMS AND SECTIONS
An item is the base element in the body section. TokenItems, textItems, breakItems and blankLineItems are all just specialized types of items.

TokenItems are special items that represent a name/value pair. A common example in SentryOne messages is “[Condition]: SQL Server: Blocking SQL”. The name (Condition) is a description of the value (SQL Server: Blocking SQL). The value for these tokenItems is set during the program execution.

MultiTokenItems are special tokenItems. They convert one or more tokenItems into a single token with specialized formatting. They have a “converter” attribute which is the name of a registered converter, and they reference one or more tokenItems.
TextItems are a special type of item that displays a literal text value. BreakItems and blankLineItems are specialized textItems with pre-set text for ease of use. The blankLineItem simply inserts a blank line in a message. The breakItem takes whatever the current section format break is (more about this below) and places that in the message.

A section element is simply a way to group the above items. Sections can also contain other sections. Nested sections are indented using the indentStyle from the current sectionStyle.

**ITEM AND SECTION ATTRIBUTES**

All items have the following attributes:

- id: A unique string identifying the item.
- visible: A boolean that determines if the item is visible in the message.
- before: The id of the item, token or section that this item should be placed before.
- after: The id of the item, token or section that this item should be placed after.
- requires: The id of an item, token or section that must be visible for this item to be visible.
- indent: A boolean that determines if the item should be indented.
- itemStyleId: The id of the itemStyle (format) that should be used for this item.

TextItems have one additional attribute:

- text: The text that will be displayed in the final message.

TokenItems have additional attributes:

- name: The token name to be displayed in the message. If none is assigned, a default name based on the item id is displayed.
- hideIfNoValue: A boolean that when set to true, hides the token if no value is assigned.
- tokenItemStyleId: The id of the tokenStyle (format) that should be used for this token.

MultiTokenItems have all of the tokenItem attributes plus an additional attribute and element:

**Attribute:**

- converter: The name of the registered converter

**Element:**

- token: An element that represents a token to be passed to the converter. It has a single attribute—id—that is the id of a valid token. MultiTokenItems may have one or more token elements.

Sections have similar attributes to items:

- id: A unique string identifying the item.
- visible: A boolean that determines if the item is visible in the message.
- before: The id of the item, token or section that this item should be placed before.
- after: The id of the item, token or section that this item should be placed after.
- requires: The id of an item, token or section that must be visible for this item to be
visible.

- **indent**: A boolean that determines if the item should be indented.
- **sectionStyleId**: The id of the sectionStyle that should be used for this section.
- **itemStyleId**: The id of the itemStyle (format) that should be used for this item.
- **tokenItemStyleId**: The id of the tokenStyle (format) that should be used for this token.

**STYLES**

Styles are a very important part of how message customization works. There are three types of styles: section styles, item styles and token styles. The styles are defined in the format section of the XML. The styles are then referenced by their IDs in the sections, items and tokens. These styles use format strings to identify how to change an item, token or section when the message is built. The format strings use keywords to represent the data so you can add formatting content around them.

For example, token styles use %itemname% to represent the name of a token and %itemvalue% to represent its value. So, to create the finished message value of "[Condition]: SQL Server: Blocking SQL", one would use the format string: "[%itemname%]: %itemvalue%". If the desired finished value was "Condition-->SQL Server: Blocking SQL" the format string would need to be changed to "%itemname%-->%itemvalue%". You can also add whitespace characters such as newline (\n\n) or tab (\t).

**STYLE ATTRIBUTES**

Section styles (sectionStyle elements) have three attributes:

- **id**: A unique string identifying them
- **indentStyle**: A format string to determine how a nested section should be indented.
  - Keyword: %sectionitem% represents each item in the section.
  - Example: " %sectionitem%" places four spaces before each item in the section.
- **formatStyle**: A format string to determine how a section should be formatted.
  - Keyword: %sectionitems% represents the entire section as a whole.
  - Example: "%sectionitems%----------\n " places a line of dashes as a break after the section.

Item styles (itemStyle elements) have two attributes:

- **id**: A unique string identifying this style.
- **indentStyle**: A format string to determine how an item should be indented (when the indent attribute is set to ‘true’).
  - Keyword: %item% represents the item.
  - Example: "\t%item%" places a tab character in front of the item.

Token styles (tokenItemStyle elements) extend item styles and have three additional attributes:

- **formatStyle**: A format string to determine how a token should be formatted.
  - Keywords: %itemname% represents the token name; %itemvalue%
represents the value of the token.
- Example: “[%itemname%]: %itemvalue%” creates the standard “[Name]: Value” token style.
- itemStyleId: The id of the itemStyle this tokenStyle references. Because a token is an item underneath, the format needs to specify the item style as well.
- itemValueStyle: A .NET format string for the item value. This string gives you the ability to format the token value itself using .NET format string which is useful for date or number formatting.
  - Example: “0.##” formats a decimal value to a maximum of two decimal places.

CUSTOMIZING A MESSAGE

SECTIONS OF THE MESSAGE EDITOR

Tokens Pane
The Tokens Pane, located in the top left corner of the Message Editor, displays a list of tokens that are available for use in the message.

Text Items Pane
The Text Items Pane contains various XML elements that can be copied and pasted into the Message Level XML pane to add custom text and line breaks to the message.

Formats Pane
The Formats Pane contains XML attributes that can be used with an element to adjust the formatting of the element.

Message Level XML Pane
The Message Level XML Pane is where you can edit the message. Changes made in this pane are reflected in the Complete XML and Message Sample Panes.

Complete XML Pane
The Complete XML Pane displays the final product of the template displayed in the Message Level XML Pane plus all of the templates that are imported by it. You can use this pane as guideline when making changes to the Message Level XML.

Message Sample Pane
The Message Sample Pane displays a mock-up of what the complete XML will become.

Conditions That Use Template Pane
This pane shows a list of conditions that are currently using the template that you’re editing. The check boxes give you the ability to quickly save a new template and apply it to multiple conditions.

EDITING A MESSAGE

Adding Additional Information
New items can be added inside or outside of a section. Available nodes are listed in the Tokens Pane and can either be copy-and-pasted, highlighted-and-dragged, or typed
directly into the editor. Set the "before" or "after" attributes to place the item in a particular spot in the message.

**Removing or Hiding an Item**

If an item is only in the Message Level XML, you can simply delete that node from the XML. If the item is in an imported template, you can hide the item in the finished message by setting the "visible" attribute to false. You can also set the "hideIfNoValue" attribute to true which will hide the item if there is no value assigned to it.

**Moving an Item**

Items are displayed in the final message as they are placed in the XML, top to bottom, unless "before" or "after" attributes are placed on the items or sections. This quickly gets confusing with the inheritance chain, so the easiest way to move an item is to set its "before" or "after" attribute.

**Adding, Hiding, or Moving a Section**

Sections are a lot like items (by design) so you can add one by typing in a new `<section> </section>`, hide one by setting its visible property to false and move it by setting its "before" or "after" attribute.

**Changing a Token Name**

You can change the displayed name of a token by setting its "name" property.

**Indenting an Item**

All items already have an inherited default indent style so you can simply set the "indent" attribute for an item to true. For custom indents, you can create a new itemStyle with your custom indent and apply that style to a body, section or item.

**Formatting a Token**

Each template will have several inherited token styles. They are listed in the Formats Pane of the Message Editor. You can copy-and-paste or highlight-and-drag these attributes into the body, section or token item. For custom formatting, you can define a new tokenItemStyle in the format section and set the formatStyle string to suit your needs. You can then reference this new style by id in the attributes of the body, section or tokenItem.

Example: `tokenItemStyle` definition to get a token like "{Condition}-->Name of Condition":

```xml
<tokenItemStyle id="NewStyle" itemStyleId="defaultItem" formatStyle="{%itemname%}-->%itemvalue%"/>
```

**Formatting a Section**

Each template will have at least one inherited section style. They are listed in the Formats Pane of the Message Editor. You can copy-and-paste or highlight-and-drag the attribute into the body or section. For custom formatting, you can define a new sectionStyle in the format section and set the formatStyle string to suit your needs. You can then reference this new style by id in the attributes of the body or section.

Example `sectionStyle` definition to replace the standard line of dashes after a section with a line of equal signs:

```xml
<sectionStyle id="NewSectionStyle" formatStyle="%sectionitems%=======================
indentStyle="%sectionitem%"/>
```
Formatting a Token Value

You can define a .NET format string to format a token value itself. This is especially useful in formatting dates and number values. Define a new tokenItemStyle in the format section, set up the "formatStyle" and "itemStyleId" attributes (or copy them from another style) and then set the "itemValueStyle" to a .NET format string.

7.3 Settings

What makes Settings especially relevant in the context of the Alerting and Response System is that certain Settings define the criteria of when a Condition is considered to be met. This includes Runtime Threshold Settings.

Other Settings, known as Source Settings define what events are collected by SentryOne. The Source Settings define minimum duration thresholds for the collection of events taking place in the monitored environment. Certain Setting control the Synchronization of events, including History Filter Settings. History Filters are available for select Event Sources. With History Filters you define rules which control exactly what types of events the Monitoring Service writes into Event History concerning your monitored servers. If an event does not meet the criteria you have defined in the History Filter, information about that event is not written to your SentryOne Database.

These Source Settings all have a direct correlation to Conditions being met, as an event must first be captured before any Condition related to it can be evaluated. These Setting include various Collection Settings.

WHERE TO CONFIGURE SETTINGS

You can configure Settings at the Global level (All Targets), Site level, Target Group level, Target level, Instance level, or at the individual object level as applicable. Settings are hierarchical, and work through the principle of inheritance. This means that Settings applied at the Global level (All Targets) are automatically inherited by all the Sites, Target Groups, Targets, Instances, and objects in your environment. Generally it is a good idea to first configure your Settings globally, and then adjust individual Settings as needed at the lower levels.

HOW TO CONFIGURE SETTINGS

To configure a Setting first select the desired node in the Navigator pane. For instance, select the All Targets node if you want to configure Settings Globally, or select an individual Instance node if you would like to configure Settings specific to just that Instance. If you do not see the Settings pane once you have selected your desired node in the Navigator pane, use the View Menu (View → Settings). Next you will want to use the drop-down lists found in the Settings pane, to select the Settings which you would like to configure.

Within the Settings pane, directly beneath the drop-down lists, the full path of each Settings group is displayed. This path contains all of the ancestor objects for the selected Setting. Each Setting also has an Inherit From Parent flag which specifies if the Setting is being inherited. When this flag is set to true the name of the ancestor object which is passing down the inherited Settings will be listed. For example, if you initially configure your Settings globally, as you drill down to the Target and Instance level, you will see that each child Setting contains an Inherit From Parent (Global): listing, with the flag set to True. This lets you know that these particular Settings are being inherited from the Global level (All Targets).
Configure Settings Example

If you wanted to configure the **Top SQL Minimum Duration Collection Setting** Globally:

1. Select the **All Targets** node in the **Navigator Pane**
2. In the **Settings pane**, use the top drop-down list and select **SQL Server Settings**.
3. Use the second drop-down list to select **Top SQL Source**. You should now see the Top SQL Source Settings that are being applied Globally.
4. Change the **Minimum Duration** to the desired value, it will be saved automatically.

If you wanted to configure the **SQL Server Agent Job Maximum Runtime Threshold Percent** for an individual Instance:

1. Select the desired **Instance** node in the **Navigator pane**.
2. In the **Settings pane**, use the top drop-down list and select **SQL Server Settings**.
3. Use the second drop-down list to select **SQL Server Agent Job**. You should now see the SQL Server Agent Job Settings that are configured for the Instance.
4. Change the **Inherit From Parent Setting** to False.
5. Change the **Maximum Runtime Threshold Percent** to the desired value, it will be saved automatically.

**COMPREHENSIVE LISTING OF SETTINGS**

Use the links below to jump to the individual topics for more information about various Settings. You will find a description of each **Setting** and a **Related Condition** column. This **Related Condition** column contains a link to the Condition(s) that are affected by the specific Settings.

- SQL Server Settings
- SSAS Settings
- Windows Settings

⚠️ For information about **Fragmentation Manager** related Settings, including Database Source Settings please see the **Fragmentation Manager** topic.

**GENERAL INFORMATION ABOUT SELECT SETTINGS**

**Runtime Thresholds Settings**

Runtime Thresholds can be set for a number of sources such as SQL Server Agent Jobs.

**Percent Based**

A value for **Minimum Runtime Threshold Percent** and **Maximum Runtime Threshold Percent** can be specified. The global Minimum Threshold Percent default is 10%. The Maximum Threshold Percent default is 250%. This means that anytime a job runs for less than 10% or longer than 250% of it's average runtime the Associated Condition would be met and any Actions configured for that Condition would also take place. If too many messages are being received, these settings can be adjusted as needed.

**Time Based**

Explicit time-based thresholds can also be specified. Anytime an explicit time-based threshold is specified it will override the percentage based thresholds for that object. Time-based thresholds are usually less valuable at the Global level, particularly the Minimum Runtime Threshold, which
doesn't have much value at all globally. Explicit runtime thresholds tend to be more applicable at the actual Instance or object level for overriding the global percentage thresholds on a case-by-case basis.

**NOTE:** One notification will be sent once the Maximum Runtime Threshold has been met and a separate notification will be sent once it has completed.

For example, consider a job that has a great deal of volatility in runtime such as a transaction log backup, which can run for anywhere between 30 seconds and 30 minutes, and it's average runtime is 5 minutes. To avoid unnecessary percentage-based threshold notifications for the job, one option would be to set its **Maximum Runtime Threshold** to "35 Minutes" and **Minimum Runtime Threshold** to "20 Seconds". This can be done by selecting either the job's node in the Navigator or an instance of the job on the calendar, then follow the same steps as above to access and change the job's runtime threshold settings.

**NOTE:** Runtime Threshold messages for Reporting Services Reports completing in less than two minutes will not be processed until they are complete.

**Performance Counter Threshold Settings**

Screen is used to set the minimum and maximum values for the counter thresholds of that particular counter. These will be used to trigger the **Performance Counter: Threshold Min** and **Max** conditions if any actions are enabled for them. This option is only available when selecting a specific performance counter from the Navigator Pane. There can be only one minimum and one maximum value for a given counter, regardless of how many objects are linked to the counter. For more information about Performance Counters see the **Performance Counter ('Monitoring Event Object Performance' in the on-line documentation) topic.**

### TARGET SETTINGS

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance Window</td>
<td>Used to set a schedule for the Maintenance Window. Only Log Actions will be triggered for Failsafe conditions inside the Maintenance Window. Maintenance Window setting allow a time frame to be specified while daily maintenance activities take place on that Instance. For more information see the <strong>Schedules and Windows</strong> topic.</td>
<td>Failsafe Conditions</td>
</tr>
</tbody>
</table>

### EVENT CHAIN SETTINGS

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Runtime Threshold</td>
<td>Define the Maximum allowed runtime for the Event Chain.</td>
<td>Event Chain: Runtime Threshold Max</td>
</tr>
</tbody>
</table>
### EVENT CHAIN NODE SETTINGS

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Runtime Threshold</td>
<td>Define the Minimum allowed runtime for the Event Chain.</td>
<td>Event Chain: Runtime Threshold Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Runtime Threshold</td>
<td>Define the Maximum allowed runtime for the Event Chain node.</td>
<td>Event Chain: Runtime Threshold Max</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum Runtime Threshold</td>
<td>Define the Minimum allowed runtime for the Event Chain node.</td>
<td>Event Chain: Runtime Threshold Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SITE SETTINGS

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>General SMTP Server</td>
<td>The SMTP server or server group used for email notifications. See also: SMTP Configuration</td>
<td>N/A</td>
</tr>
</tbody>
</table>

7.3.1 SQL Server Settings

### SQL SERVER SETTINGS

#### Blocking SQL

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Duration Threshold</td>
<td>Set the Maximum allowed Duration for the object.</td>
<td>Blocking SQL: DurationThreshold Max</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Blocking SQL Source
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows To Synchronize</td>
<td>Set the maximum rows of historical event data to collect from this source.</td>
<td>N/A</td>
</tr>
<tr>
<td>Minimum Block Duration</td>
<td>Set the time that must elapse before a Block Condition will be triggered.</td>
<td>All Blocking SQL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Conditions</td>
</tr>
</tbody>
</table>

**Databases Source (See [Fragmentation Manager](#) section for additional details)**

**Deadlocks Source**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows To Synchronize</td>
<td>Set the maximum rows of historical event data to collect from this source.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>The filter used for history synchronization. Only events that meet the filter will be written to the database. See also: <a href="#">History Filter</a></td>
<td></td>
</tr>
</tbody>
</table>

**History Filter**

**: Note:** The Deadlocks History Filter only impacts what is displayed on the Event Calendar and does not impact deadlock collection. For more information see [Synchronization of Performance Analysis sources](#) section of the History filter topic.

| Collect Deadlock Events       | Specifies whether deadlock statement collection is enabled.                | All Deadlock SQL  |
|                               |                                                                             | Conditions        |

**Maintenance Plan Source**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-watch</td>
<td></td>
<td>All Maintenance Plan Conditions</td>
</tr>
<tr>
<td>New</td>
<td>Specify whether or not new objects will be set to watched when they are discovered.</td>
<td></td>
</tr>
<tr>
<td>Objects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Rows To Synchronize</td>
<td>The maximum rows of historical data to collect for this source.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**SentryOne User Guide | 121**

**Reporting Services Report**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Runtime Threshold</td>
<td>Define the maximum allowed runtime for the object.</td>
<td>Reporting Services Report: Runtime Threshold Max</td>
</tr>
<tr>
<td>Maximum Runtime Threshold Percent</td>
<td>Define the maximum allowed runtime percent for the object.</td>
<td>Reporting Services Report: Runtime Threshold Max</td>
</tr>
</tbody>
</table>
### Minimum Runtime Threshold
Define the minimum allowed runtime for the object.

### Minimum Runtime Threshold Percent
Define the minimum allowed runtime percent for the object.

### Reporting Services Report Source

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-watch New Objects</td>
<td>Specify whether or not new objects will be set to watched when they are discovered.</td>
<td>All Reporting Services Report Conditions - An object must be watched in order for Conditions to be evaluated for it.</td>
</tr>
<tr>
<td>Maximum Rows To Synchronize</td>
<td>The maximum rows of historical data to collect for this source.</td>
<td>N/A</td>
</tr>
<tr>
<td>History Filter</td>
<td>The Filter used for history synchronization. Only events that meet the filter will be written to the database. See also: History Filter</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### SQL Server Agent Alerts Source

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-watch New Objects</td>
<td>Specify whether or not new objects will be set to watched when they are discovered.</td>
<td>All SQL Server Agent Alert Conditions - An object must be watched in order for Conditions to be evaluated for it.</td>
</tr>
<tr>
<td>Maximum Rows To Synchronize</td>
<td>The maximum rows of historical data to collect for this source.</td>
<td>N/A</td>
</tr>
<tr>
<td>History Filter</td>
<td>The Filter used for history synchronization. Only events that meet the filter will be written to the database. See also: History Filter</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### SQL Server Agent Job

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Runtime Threshold</td>
<td>The maximum allowed runtime for the object.</td>
<td>SQL Server Agent Job: Runtime Threshold Max</td>
</tr>
<tr>
<td>Maximum Runtime Threshold Percent</td>
<td>The maximum allowed runtime percentage for the object.</td>
<td>SQL Server Agent Job: Runtime Threshold Max</td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
<td>Related Condition</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Minimum Runtime Threshold</td>
<td>The minimum allowed runtime for the object.</td>
<td>SQL Server Agent Job: Runtime Threshold Min</td>
</tr>
<tr>
<td>Minimum Runtime Threshold Percent</td>
<td>The minimum allowed runtime percent for the object.</td>
<td>SQL Server Agent Job: Runtime Threshold Min</td>
</tr>
<tr>
<td>Queue Others for up to</td>
<td>The maximum length of time this job will queue other jobs. See Job Queuing for more information.</td>
<td>N/A</td>
</tr>
<tr>
<td>Auto-Start Threshold</td>
<td>Any job whose next scheduled run time is beyond the specified threshold will be started automatically upon leaving the queue. If a job’s next scheduled run time is before the threshold it will not auto-start.</td>
<td>N/A</td>
</tr>
<tr>
<td>Queue Type</td>
<td>The behavior when this job is queued. See Job Queuing for more information.</td>
<td>SQL Server Agent Job: Queued</td>
</tr>
<tr>
<td>Queue For Up To</td>
<td>The maximum length of time this job can queued. See Job Queuing for more information.</td>
<td>N/A</td>
</tr>
<tr>
<td>Auto-Start Type</td>
<td>Controls the auto-start behavior for the job when it leaves the queue. “Use default setting” will cause the queuing job’s &quot;Auto-Start Threshold&quot; setting to be used.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**SQL Server Agent Jobs Source**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-watch New Objects</td>
<td>Specify whether or not new objects will be set to watched when they are discovered.</td>
<td>All SQL Server Agent Job Conditions - An object must be watched in order for Conditions to be evaluated for it.</td>
</tr>
<tr>
<td>Missed Run Threshold</td>
<td>The length of time that must pass before an object run is considered to be missed.</td>
<td>SQL Server Agent Job: Run Missed</td>
</tr>
<tr>
<td>Maximum Rows To Synchronize</td>
<td>The maximum rows of historical data to collect for this source.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**SQL Server Agent Log Source**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-watch New Objects</td>
<td>Specify whether or not new objects will be set to watched when they are discovered.</td>
<td>All SQL Server Agent Log Source Conditions - An object must be watched in order for Conditions to be evaluated for it.</td>
</tr>
<tr>
<td>Maximum Rows To Synchronize</td>
<td>The maximum rows of historical data to collect for this source.</td>
<td>N/A</td>
</tr>
<tr>
<td>History Filter</td>
<td>The Filter used for history synchronization. Only events that meet the filter will be written to the database. See also: History Filter</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### SQL Server Instance

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-enable SQL Server Agent Tokens</td>
<td>On SQL Server 2005 and above, SQL Agent Tokens are disabled as a security measure. They must be enabled to allow SentryOne to watch Alerts.</td>
<td>All SQL Server Agent Alert Conditions</td>
</tr>
<tr>
<td>Auto-recycle large SQL Server Agent Logs</td>
<td>Whether or not to automatically recycle large Agent Log files. SentryOne polls the Agent for history synchronization. If an Agent Log file becomes too large, this polling will cause timeout problems. To prevent this, SentryOne will stop polling and issue a Client Alert if the agent log file exceeds 1MB. This problem often manifests when informational level logging is enabled.</td>
<td>N/A</td>
</tr>
<tr>
<td>Maximum Queue Length</td>
<td>The maximum number of objects that can be queued at one time. This setting always applies to the Instance level.</td>
<td>SQL Server Agent Job: Queued</td>
</tr>
<tr>
<td>Synchronize Type</td>
<td>This setting refers to how the SentryOne Client will respond if it discovers the data in the SentryOne Database is not up to date. The default is to show a message box warning of the condition and asking if the SentryOne Client should synchronize the information itself.</td>
<td>N/A</td>
</tr>
<tr>
<td>Synchronization Threshold</td>
<td>The amount of time that must pass since the last successful synchronization by the SentryOne Monitoring Service before the Client attempts to synchronize directly with the target.</td>
<td>N/A</td>
</tr>
<tr>
<td>Max Databases to Synchronize (largest first)</td>
<td>The maximum number of databases to synchronize, ordered by total size. This includes system databases.</td>
<td>N/A</td>
</tr>
<tr>
<td>Max User Database Files to Synchronize (most active first)</td>
<td>The maximum number of user database files to synchronize, ordered by highest activity. Activity is defined by number of bytes read and written. This number excludes system databases which are automatically included in the synchronization.</td>
<td>N/A</td>
</tr>
<tr>
<td>Maintenance Window</td>
<td>Used to set a schedule for the Maintenance Window. Only Log Actions will be triggered for Failsafe conditions inside the Maintenance Window. Maintenance Window setting allow a time frame to be specified while daily maintenance activities take place on that Instance. For more information see the Schedules and Windows topic.</td>
<td>Failsafe Conditions</td>
</tr>
</tbody>
</table>

### Top SQL

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Runtime Threshold</td>
<td>The maximum allowed runtime for the object. This default value is 1 hour.</td>
<td>Top SQL: Runtime Threshold Max</td>
</tr>
</tbody>
</table>

### Top SQL Source
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows to Synchronize</td>
<td>The maximum rows of historical data to collect for this source.</td>
<td>N/A</td>
</tr>
<tr>
<td>Minimum Duration</td>
<td>The Minimum Duration setting determines the required duration for a Top SQL event to be collected. The default setting is 5 seconds. For safety reasons, it cannot be reduced below 100ms unless one of the CPU, Reads or Writes filters is &gt;0.</td>
<td>All Top SQL Conditions</td>
</tr>
<tr>
<td>Minimum CPU</td>
<td>The minimum required CPU for a statement to be logged.*</td>
<td>All Top SQL Conditions</td>
</tr>
<tr>
<td>Minimum Reads</td>
<td>The minimum required reads for a statement to be logged.*</td>
<td>All Top SQL Conditions</td>
</tr>
<tr>
<td>Minimum Writes</td>
<td>The minimum required writes for a statement to be logged.*</td>
<td>All Top SQL Conditions</td>
</tr>
<tr>
<td>Collect Trace Events</td>
<td>Specify whether or not to collect Trace Events. If Collect Trace Events is disabled, SentryOne will not start a trace or collect any Top SQL events from a monitored target.</td>
<td>N/A</td>
</tr>
<tr>
<td>Collect Statement Events</td>
<td>Specify whether or not to collect Statement Events. If Collect Statement Events is disabled, statement-level information will not be available in the Plan Explorer Statements Tree when viewing a query plan.</td>
<td>N/A</td>
</tr>
<tr>
<td>Collect Execution Plans</td>
<td>Specify whether or not to collect Execution Plans. If Collect Execution Plans is disabled, Top SQL event collection will occur as normal, however query plan information will not be automatically collected for those events. Plan information can still be collected on demand from within a Plan Explorer session, however.</td>
<td>N/A</td>
</tr>
<tr>
<td>Max Plan Size to Collect (MB)</td>
<td>Specifies the max plan size to collect in MB. Default value is 3.</td>
<td>N/A</td>
</tr>
<tr>
<td>Running Events Polling Interval</td>
<td>Specifies how often the service checks for running events.</td>
<td>N/A</td>
</tr>
<tr>
<td>Collect Query Stats</td>
<td>Specifies whether query stats collection is enabled. This setting must be set to True for Procedure Stats and Query Stats to function.</td>
<td>N/A</td>
</tr>
<tr>
<td>Query Stats Sample Interval</td>
<td>Specifies how often to sample query stats.</td>
<td>N/A</td>
</tr>
<tr>
<td>Filter Time Span</td>
<td>Specifies the base length of time over which the collection filters will be applied to Query Stats.</td>
<td>N/A</td>
</tr>
<tr>
<td>Filter Factor</td>
<td>The Filter Factor is calculated by dividing the Query Stats Sample Interval by the Filter Time Span. The</td>
<td>N/A</td>
</tr>
</tbody>
</table>
collection filters such as Minimum Duration are multiplied by this value when applied to Query Stats collection.

**Top SQL Source Note:**

There is an **AND** relationship that exists between the *Minimum Duration*, *Minimum CPU*, *Minimum Reads*, and the *Minimum Writes* Collection Settings. This means, that in order to be collected as Top SQL, the event will need to satisfy each individual Collection Setting. For example, if you set the Minimum Duration at 10 seconds and the Minimum Reads at 25, an event would need to meet both a Minimum Duration of 10 seconds **AND** a Minimum Reads of 25 to be captured in Top SQL.

*Minimum Duration* cannot be set below 100ms unless *Minimum CPU*, *Minimum Reads*, or *Minimum Writes* is greater than 0. This lower limit is enforced because setting this thresholds below 100ms for an extended period of time could dramatically increase the volume of data collected and stored by SentryOne, and have a negative impact on the monitored server. SentryOne's *Quick Trace* functionality is better suited to analyze extremely short duration events.

### 7.3.2 SSAS Settings

**SQL SERVER ANALYSIS SERVICES SETTINGS**

**Analysis Services Instance**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect Memory by Category Data</td>
<td>Enable or disable Memory by Category Data. This setting is used to provide more detail than the basic Cleaner memory data, but is higher overhead to collect. Use caution when enabling to ensure it does not impact performance. The default value for this setting is False.</td>
<td>N/A</td>
</tr>
<tr>
<td>Synchronize Type</td>
<td>This setting refers to how the SentryOne Client will respond if it discovers the data in the SentryOne Database is not up to date. The default is to show a message box warning of the condition and asking if the SentryOne Client should synchronize the information itself.</td>
<td>N/A</td>
</tr>
<tr>
<td>Synchronization Threshold</td>
<td>The amount of time that must pass since the last successful synchronization by the SentryOne Monitoring Service before the Client attempts to synchronize directly with the target.</td>
<td>N/A</td>
</tr>
<tr>
<td>Maintenance Window</td>
<td>Used to set a schedule for the Maintenance Window. Only Log Actions will be triggered for Failsafe conditions inside the Maintenance Window. Maintenance Window setting allow a time frame to be specified while daily maintenance activities take place on that Instance.</td>
<td>Failsafe Conditions</td>
</tr>
</tbody>
</table>
For more information see the [Schedules and Windows](#) topic.

### Top Commands

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Runtime Threshold</td>
<td>Define the maximum allowed runtime for the object.</td>
<td>Top Commands: Runtime Threshold Max</td>
</tr>
</tbody>
</table>

### Top Command Source

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows to Synchronize.</td>
<td>The maximum rows of historical event data to collect from this source.</td>
<td>N/A</td>
</tr>
<tr>
<td>History Filter</td>
<td>The filter is used for history synchronization. Only events that meet the filter will be written to the database. See also: History Filter</td>
<td>N/A</td>
</tr>
<tr>
<td>Minimum Duration</td>
<td>Set the minimum required duration for a statement to be logged.</td>
<td>All Top Commands Conditions</td>
</tr>
<tr>
<td>Collect MDX/DMX/XMLA Events</td>
<td>Specifies whether command collection is enabled.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### 7.3.3 Windows Settings

**WINDOWS SETTINGS**

**Windows Instance**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Type</td>
<td>This setting refers to how the SentryOne Client will respond if it discovers the data in the SentryOne Database is not up to date. The default is to show a message box warning of the condition and asking if the SentryOne Client should synchronize the information itself.</td>
<td>N/A</td>
</tr>
<tr>
<td>Synchronization Threshold</td>
<td>The amount of time that must pass since the last successful synchronization by the SentryOne Monitoring Service before the Client attempts to synchronize directly with the target.</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Maintenance Window

Used to set a schedule for the Maintenance Window. Only Log Actions will be triggered for Failsafe conditions inside the Maintenance Window. Maintenance Window setting allow a time frame to be specified while daily maintenance activities take place on that Instance.

For more information see the [Schedules and Windows](#) topic.

### Collect Processes

Specifies whether process collection is enabled.

### Uncategorized Process Filter

The filter controls which uncategorized processes are collected. Categorized or "well-known" processes are always collected. See Also: [Processes](#)

### Windows Event Logs Source

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-watch New Objects</td>
<td>Specify whether or not new objects will be set to watched when they are discovered.</td>
<td></td>
</tr>
<tr>
<td>Maximum Rows to Synchronize</td>
<td>The maximum rows of historical data to collect for this source.</td>
<td>N/A</td>
</tr>
<tr>
<td>History Filter</td>
<td>See <a href="#">History Filter</a> and <a href="#">Windows Event Logs</a> source topic.</td>
<td></td>
</tr>
</tbody>
</table>

### Windows Task

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Related Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Runtime Threshold</td>
<td>The maximum allowed runtime for the object.</td>
<td>Windows Task: Runtime Threshold Max</td>
</tr>
<tr>
<td>Maximum Runtime Threshold Percent</td>
<td>The maximum allowed runtime percentage for the object.</td>
<td>Windows Task: Runtime Threshold Max</td>
</tr>
<tr>
<td>Minimum Runtime Threshold</td>
<td>The minimum allowed runtime for the object.</td>
<td>Windows Task: Runtime Threshold Min</td>
</tr>
</tbody>
</table>
### 7.4 Response Rulesets

**Response Rulesets** control how often Actions are taken in response to Conditions. The default ruleset is **Notify Every Time**, meaning that the Action will be taken every time the Condition is met. **Response Rulesets** allow for an extra level of configurability for certain notifications. They do this 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 5 minutes before an email is to be sent. Additionally, you can specify under what 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 will be 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](#).

💡 **Response Rulesets** can be a powerful tool to help you fine tune alerts. **Response Rulesets** are available globally, meaning the same Response Ruleset can be applied to multiple Conditions. Keep in mind that any changes to a Response Ruleset will affect every Condition where the **Response Ruleset** has been applied.

Windows are also available to help you fine tune alerting in your environment. **Windows** may be applied directly to any configured Action to control the time frame of when that Action will be processed. You may 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**.
Ruleset Type | Description
---|---
**Count Based** | With **Count Based** Rulesets you can control:
- How many times a Condition must occur within a specified time frame before Actions are taken.
- How many times a Condition must reoccur before subsequent Actions are taken.
- How long to continue taking action when a Condition is occurring.

See the Response Ruleset [Count Based example](#) for more information.

**Time Based** | With **Time Based** Rulesets you can control:
- How long a Condition must exist before actions are taken.
- How much time must elapse before subsequent actions are taken.
- How long to continue taking action when a Condition is occurring.

See the Response Ruleset [Time Based example](#) for more information.

---

**RESPONSE RULESET EXAMPLES**

For example use scenarios see the [Response Ruleset Examples](#) topic.

**RULESET MANAGEMENT**

**Response Rulesets** can be managed using the **Tools menu → Manage Response Ruleset** command. In this window you can create, edit, and delete rulesets. You can also assign rulesets to all of the 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 **Action Pane**.

**CREATE A NEW RESPONSE RULESET**

To create a new Response Ruleset from within the Manage Response Ruleset dialogue select the **New Ruleset** button. Choose a **name** for the Response Ruleset. Next use the **Ruleset Type** box to select either a Count based or Time Based ruleset. Depending on which Ruleset type you are creating the configuration options will be as follows.

**Ruleset Configuration Options**
Count Based Configuration Options

**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.
- Within (not checked) – ignore time limit, enable subsequent actions.

**Process Subsequent Actions Every:**
- Occurrences – 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.
ASSIGN A RESPONSE RULESET TO AN ACTION

You can assign a Response Ruleset to an Action from within the Manage Response Ruleset dialogue as follows.

1. Open the Manage Response Ruleset dialogue. (Tools menu → Manage Response Ruleset)
2. Select any Action found within the Configured Actions section.
3. Select the drop-down arrow in the Ruleset column to see Response Rulesets applicable for the Action.
4. Click on the Response Ruleset that you would like to apply.

You can also assign a Response Ruleset to an Action from the Response Ruleset tab found within the Action Pane.

1. In the Navigator Pane select the node appropriate to the level that you would like to configure the Action for.
2. Select the Action that you would like to apply a ruleset for in the Conditions pane. (View menu → General/Failsafe/Audit Actions)
3. Select the Rulesets tab found directly beneath the Action.
4. Click on the Response Ruleset that you would like to apply.

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.

7.4.1 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 will be taken every time the Condition is met. Response Rulesets allow for an extra level of configurability for certain notifications. They do this 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 with Count Based Rulesets you can control:

- How many times a Condition must occur within a specified time frame before Actions are taken.
- How many times a Condition must reoccur 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 is 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 there is a problem.

To prevent getting an alert on every failure, you could 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 was only sent after the job failed multiple times within a specified time frame. For the example we will do this by configuring a Count Based Response Ruleset that will require the job to fail 3 times within 5 minutes.

To accomplish this you need to apply a Count Based Response Ruleset at the job level (FrequentJob). You will want to 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.

First you will want to 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. Next open the General Actions section in the Conditions pane. If you do not see the Conditions pane use the View menu -> Conditions.
3. Click the Add button found in the Explicit section of the General Actions section. This will open the Select Action window.
4. Expand the SQL Server Agent Job: Failure Condition. Use the check box to select the Send Email Action. Click the OK button to add the configured action. Be sure to specify a Selected Target in the Action Settings tab.

Next you will want to apply a Response Ruleset to the configured Action.

1. Select the configured email action in the Explicit section of the Conditions pane.
2. Select the Response Rulesets tab found directly beneath the Explicit section of the Conditions pane. You should see that the default Response Ruleset, Notify Every Time is applied. Click the New button to open the Response Ruleset Editor.
3. Give the Response Ruleset a name, remembering that it is a best practice to name a custom ruleset by the object and Condition that it applies to.
4. Fill in the information as specified above and click the Save button 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 will only be sent alerting you that the FrequentJob failed, if it does so 3 times within 5 minutes.

Only the SQL Agent Job FrequentJob will be impacted by this change. Other Agent Jobs would still be subject to the Send Email Action configured at the Instance level which has the default Response Ruleset of Notify Every Time.

💡 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 would be 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:**

With Time Based Rulesets you can control:

- How long a Condition must exist before Actions are taken.
- How much time must elapse 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 are being alerted by email that these Instances are going Offline and Online, but when you check these Instances these seem to be false notifications. This can happen if there are 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 you could set a Response Ruleset to limit the processing of the Send Email Action for the SQL Server Instance: Offline Condition. This Response Ruleset would add additional criteria to the processing of the Send Email Action so that an email is only sent after the Instance was detected to be in an offline state for a specific period of time.

For this example we will do this by:
1. Configuring a **Time Based Response Ruleset** that will require the *SQL Server: Offline Condition* to be true for 5 minutes.

2. Disabling the **SQL Server Instance: Online Condition**.

To accomplish this you would need to apply a **Time Based Ruleset** at the *Site level* for the **TransientSite**. You will want to 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.

First you will want to 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**.

2. Next Open the **Failsafe Actions section** in the **Conditions pane**. If you do not see the Conditions pane use the **View menu -> Conditions**.

3. Click the **Add button** found in the Explicit section of the Failsafe Actions section. This will open the Select Action window.

4. Expand the **SQL Server Instance: Offline Condition**. Use the check box to select the **Send Email Action**. Click the **OK button** to add the configured action. Be sure to specify a **Selected Target** in the Actions Settings tab.

Next you will want to apply a **Response Ruleset** to the configured Action.

1. Select the configured email action in the Explicit section of the Conditions pane.

2. Select the **Response Ruleset tab** found directly beneath the Explicit section of the Conditions pane. You should see that the default Response Ruleset, *Notify Every Time* is applied. Click the **New button** to open the Response Ruleset Editor.

3. Give the Response Ruleset a name, remembering that it is a best practice to name custom ruleset by the object and condition that it applies to.

4. Fill in the information as specified above and click the **Save button** 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 will only be sent alerting you that one of the Instances belonging to the TransientSite is offline, if it is detected to be in that state for longer than 5
Next we will disable the Send Email Action for the SQL Server Instance: Online Condition at the TransientSite level. This will stop any SQL Server online email notifications. We do this because with the above Response Ruleset applied, once you stop receiving emails for the SQL Server Instance: Offline Condition you will know that 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.
2. Next open the Failsafe Actions section in the Conditions pane. If you do not see the Conditions pane use the View Menu -> Conditions.
3. Expand the SQL Server Instance: Online Condition. Use the check box to select the Send Email Action. Click the OK button 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.

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 you will always have 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.

7.5 Output Content Match

The Output Content Match conditions allow you to assign an action based on a match of the output text of an event object. This feature allows you to watch for specialized conditions that may not otherwise be detected. This may prove useful when handling exit codes from Windows Tasks.

Condition Setting filters introduced in v7.2 offer a greater degree of flexibility in managing Alerts.

SETTING UP ACTIONS FOR THE OUTPUT CONTENT MATCH CONDITION

Output Content Match Actions, like all other General Actions, can be set up from the global level to the individual event object level. After selecting the appropriate node on the Navigator pane, select Add in the Conditions pane. This will bring up the Actions Selector dialog. Expand the desired Object and Output Content Match Condition. Next select the desired Action to be completed when the Condition is met, and select OK on the Actions Selector. Output Content Match options can be accessed in the Condition Settings tab.
Clicking the **Edit** button in the Condition Settings tab will bring up the **Content Match Criteria** form. From here you can specify all conditions the output text from the object has to meet before responding with the desired action. The matched text can either be in the form of a string literal, or you can enter a regular expression. To use regular expressions, prefix the match string with "regex:,", i.e. "regex:[a-z]*" to match any character a-z. Click **OK** and the Action is enabled.

**NOTE:** The Step Name listbox is only available when setting match criteria for an individual SQL Agent Job.

### CONDITION INHERITANCE

All Output Content Match criteria can be set at the global level and will be inherited like any other setting. It is important to note, however, that any match criteria defined at a lower level will effectively override the inherited criteria. Even if there is not a match at that level, SentryOne will not look further up the inheritance chain to see if there are any other match criteria set at a higher level. Inherited actions, however, will remain enabled.

For example, a Send Email action is configured at the global level for the SQL Agent Job:Output Content Match condition when the text string "overflow" appears in any job's output text. You then click on your Data Import job on Server A and enable a Log to Database action for the SQL Agent Job:Output Content Match condition and change the match criteria for this job to "insufficient". From now on, the only text string that will be checked when the Data Import job runs is "insufficient". Even if that string is not found it will not look for the "overflow" string set at the global level. However, if the "insufficient" string is matched for that job's output, both the Send Email and Log to Database actions will be taken.

When you override the Output Content Match settings at a lower level, you will be prompted if you want use the higher level Output Content Match criteria as the baseline for this setting. This feature was added because often when overriding the Output Content Match criteria, only minor changes are required, so this feature can save you a lot of work when many content match strings are involved.
OPENING THE EVENT CALENDAR

The SentryOne Event Calendar can be opened by selecting (Open → Event Calendar) from the context menu of any applicable node in the Navigator pane. It can also be opened by double-clicking the Instance and choosing Event Calendar from the product selection form.

HISTORY FILTER

- History Filter
  - Filter Editor

EVENT MANAGER VIEWS

- The Calendar View is the default view displayed in the Workspace Area.
- The List View displays all event data in a list format.

PERFORMANCE MONITORING WITH EVENTS

The following topics cover performance monitoring with SentryOne Events.

- Custom Event Views
  - Working With Custom Event Views
  - Filtering Process
  - Filter Process Flowchart
  - Leveling Backups Across a SAN or NAS

CUSTOM EVENT VIEWS

- Custom Event Views
  - Working With Custom Event Views
  - Filtering Process
  - Filter Process Flowchart
  - Leveling Backups Across a SAN or NAS

CHAINING AND QUEUING

- Chaining and Queuing
  - Event Chains
  - Job Queuing

8.1 History Filter

SentryOne provides advanced source history filtering for your monitored Targets and Instances. Source History Filter Settings are available in the synchronization group for select Event Sources (View → Settings).

With the History Filter you define rules which control exactly what types of events the Monitoring
Service writes into Event History concerning your monitored servers. If an event does not meet the criteria you have defined in the History Filter, information about that event is not written to your SentryOne Database.

Using the **History Filter** you may choose to ignore certain types of events, as you can filter on multiple columns including **Duration**, **Message Text**, and **Run Status**. The filter was designed with flexibility in mind. You may define a filter as simple as Event Duration being greater than 5 seconds, or you may choose to create complex filter conditions, using conditional groups and multiple logical operators.

Depending on your environment and the type of information you collect, using a **History Filter** with the various Event Sources can also decrease the size of your SentryOne Database. On busy servers, applying a History Filter tailored for those specific environments can also significantly reduce the amount of noise, leaving you with a smaller set of actionable information. One powerful application of the source History Filter is that it can be used with the Windows Event Logs source. The **Windows Event Logs source** is available when you are monitoring your computer with Win Sentry.

If you are looking for more control over alerts, keep in mind **Condition Settings**, which give you granular control over what you are being alerted about within your environment.

### CONFIGURING HISTORY FILTERS

The Source **History Filter** Settings are available in the synchronization group for select Event Sources (**View → Settings**). Please see the **Filter Editor topic** for details about using the Filter Editor, including context menu information.

### SYNCHRONIZATION OF PERFORMANCE ANALYSIS SOURCES

It is important to understand that the **History Filter** synchronization settings for **Deadlocks** affect only what is displayed on the Event Calendar and does not impact what is actually captured or displayed within **SQL Sentry**.

This is because the synchronization of the **Deadlocks** source uses a different process than those sources which are strictly Event Calendar only. This data is first written to various Performance Analysis specific tables within the SentryOne Database, and is then used to populate the Performance Analysis **Deadlocks tab**.

This data is then correlated and written into Event Calendar tables according to your defined synchronization settings, including any filters you have specified with the **History Filter**. This Event History data is used to populate the Event Calendar, showing you **Deadlocks** alongside your regularly scheduled Agent Jobs and various other event objects.

If you are looking for more control over alerts related to Performance Analysis sources, keep in mind **Condition Settings**, which give you granular control over what you are being alerted about within your environment. Condition Settings allow you to build complex alert criteria for Conditions, including those related to **Top SQL, Blocking SQL, and Deadlocks**.

8.1.1 **Filter Editor**

SentryOne allows you to build complex filters for the purposes of **History Filtering** and defining **Condition Settings**. This topic covers the functionality of the **Filter Editor**, which is
common to both **History Filtering** and **Condition Settings**.

**Note:** The Filter Editor is case-insensitive for all string comparisons.

**FILTER EDITOR OVERVIEW**

The below diagram identifies the various aspects of the Filter Editor.

**FILTER RULES**

Using the Filter Editor, you build criteria that an event must meet by defining rules.

Each **filter rule** is made up of three distinct parts.

1. First, you define the scope of the rule, by specifying a **column name**.
2. Secondly, you select an **operator** (criteria operator) which defines an instruction for testing the value (operand value).
3. Finally, you enter the **value** (operand value) for this filter rule.

**RULE GROUPS**

Individual **rules** are logically grouped into **rule groups**. Each group contains a logical operator which defines how the rules are combined within the group. Upon creation each group will use the **And** logical operator. When a group is created with the **And** logical operator each **filter rule** that is part of the group must be true in order for the group to evaluate as true. You may change the logical operator defined for a group by first clicking it; a drop down menu will appear, allowing you to choose an alternative (**And**, **Or**, **Not And**, **Not Or**).

**CONTEXT MENUS**

When you **click on any logical operator** additional options are available from the drop-down menu.

- The **Add Condition** command will add a new filter rule to the condition group.
- The **Add Group** command will create a new sub-group beneath the existing group with the default **And** logical operator.
- The **Remove Group** command will delete the selected group.
- The **Clear All** command, available from the root logical operator, will delete all filter rules.

Selecting the **plus symbol** next to any group’s logical operator will add a new **filter rule** to the group. Selecting the **symbol** next to any **filter rule** will delete that rule from the group.
The Calendar View is the default display in the Workspace Area. It combines both historical and future event data to present an accurate visual representation of the schedule. Visual indicators highlight overlapping active event instances, event status, and duration. Additional detail information is available with a single click via popup windows. You can use drag-and-drop to reschedule future jobs or tasks in order to “level” a schedule, thus minimizing contention between multiple events fighting for the same resources.

To navigate to the Calendar View, double-click on any root, group, Instance, source, object “Event View”, or Custom Event View node in the Navigator Pane. The actual event instances displayed on the view will vary depending on the node selected. For example if you select the Jobs node under a SQL Server, only jobs will be displayed; all other events (maintenance plans, etc.) will be filtered from the view, and only the Jobs tab will be available in the corresponding Event View filter.

RECURRING EVENTS PANE

This pane is used to display events that run frequently or long enough that they would clutter the calendar in the current interval view. Exactly which events are displayed in the recurring events pane depends on the active interval and the schedule frequency of the respective events. For example, future instances of a job that runs every 5 minutes will be pinned to the recurring events view when in a 24 hour interval, but will be displayed on the calendar in 1 hour view. To determine how long events must run before showing in this window, go to the Calendar tab in the User Preferences menu.

NOTE: Events listed in the recurring pane will not register as conflicts with other events on the Calendar.

CALENDAR CONTEXT MENUS

A Context Menu is available for all event instances by right-clicking the instance in the calendar. Exactly which context items are available depend on the source type of the selected instance.

NAVIGATION CONTROL

The navigation toggle button at the top left of the calendar allows you to move quickly forwards and backwards through the calendar.

Click the Up ▲ or Down ▼ Arrows to move to the previous or next time interval . The Right ► or Left ◄ Arrows move forward or backward one day, keeping the same interval and start time. The Plus + and Minus - buttons are used to zoom in or out in the Time Ruler, one interval at a time, keeping the interval start time the same.

DATE BAR
This bar shows the day of the week and date for each day displayed in the current view.

When in a multi-day view, if you single-click any date on the date bar it will make that date active, highlighting the date. You can zoom into a single date by double-clicking the date on the date bar. You can drill into any interval for one or more days by double-clicking on the Time Ruler at the left of the calendar (see below).

TIME RULER
On the left side of the calendar, a vertical Time Ruler shows the current time-range for the selected interval. By default, Hours are displayed. You can use the Time Ruler to zoom in or out to a desired time interval display.

When you hover over the Time Ruler, the ruler highlights a time-range. Double-click to zoom into the time-range. This is similar to clicking the Zoom In button, except that the Zoom In button does not change the start time.

CURRENT TIME
The current time is displayed (when in view) with a horizontal red bar. The bar will move as the time changes.

CURRENT TIME COLUMN
To see the exact time, mouse-over the narrow column between the time ruler and the calendar or between adjacent days.

ZOOM IN/OUT
You can also use the calendar to zoom in or out to any time intervals. Right-click anywhere there is empty space on the calendar or on the Time Ruler to bring up the context menu. Select a time interval to zoom in and out.

INTERVAL BUTTONS
Hover over the very top or bottom of the time ruler and popup arrows are displayed which you can use to quickly move to the previous or the next interval, just as with the Up ▲ or Down ▼ navigation control arrows.

EVENT INSTANCES ON THE CALENDAR
All event instances on the calendar appear with a rectangular block. Future instances don't include a Run Status indicator bar.

Step Details are accessible by clicking the bottom-right corner of the event instance. By default an event instance background is white.

Default Run Status colors are:

- Green - Successful
• Yellow - Warning
• Red - Failed
• Orange - Running
• Black - Stopped

GLYPHS
Glyphs are used to provide quick visual cues about event instances on the calendar, including their event source type, queuing settings and status, block status, etc. Each instance contains at least one glyph representative of its type, which matches exactly the icon shown in the navigator pane for the source object. Below is a calendar showing several different types of event instances.

Other informational glyphs are listed below:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Job actively blocked or blocking</td>
</tr>
<tr>
<td>↓</td>
<td>Job actively being queued</td>
</tr>
<tr>
<td>✗</td>
<td>Job cannot be queued</td>
</tr>
<tr>
<td>🔄</td>
<td>Job set to queue</td>
</tr>
<tr>
<td>🔥</td>
<td>Job actively queuing</td>
</tr>
<tr>
<td>🎨</td>
<td>Event instance or event object contains a note</td>
</tr>
<tr>
<td>⌚</td>
<td>Event instance was run as part of an event chain</td>
</tr>
<tr>
<td>🔍</td>
<td>Top SQL indicator</td>
</tr>
<tr>
<td>🟣</td>
<td>Analysis Services Top Commands indicator</td>
</tr>
<tr>
<td>🕒</td>
<td>Deadlock indicator</td>
</tr>
</tbody>
</table>

POPUP WINDOWS
To view details for any event instance, left-click it anywhere to the right of the duration bar to select it and bring up a popup window. The popup reveals details about the instance, including the last step output, start and end time, duration, runtime statistics, schedules, queuing settings, notes, etc. The actual data contained in the popup will vary based upon the instances event source type. You can clear the window by moving your mouse away from the instance.

HIGHLIGHTING RELATED EVENTS
The calendar provides several options for grouping events on the calendar. Right-clicking an event and selecting “Highlight” will display the applicable highlighting options for the event. Once the highlight option(s) are chosen, similar events to the one selected will be highlighted on the calendar. The selected elements will be persisted for all identical event sources across the installation. Using the Jump To Dashboard feature will overlay each of the selected events on the Performance Analysis Dashboard.

EVENT CONFLICTS
A conflict is defined as two active event instances running on the same server at the same time. When an instance’s runtime (as indicated by the duration bar) overlaps with one or more other instances in the same view, it is considered conflicting and its background color will be orange as shown below, otherwise the background will be white.
You can easily resolve most conflicts by dragging-and-dropping future instances on the calendar. The amount of time instances must overlap before showing as in conflict can be adjusted in User Preferences.

**NOTE:** Events listed in the recurring window will not register as conflicts with other events on the Calendar.

**TOO MUCH DATA INDICATORS**
On very busy schedules, if too many event instances exist within a given range to display all of them in a meaningful fashion, a small hashed box will be shown on the right side of the pane. Hovering over the box will show a tooltip with the number of additional events in that time period. Double-clicking the box will zoom in so each actual event can be seen. If conflicts exist within the specified range, the box will be displayed in orange (default). A filter can also be applied to reduce the number of event instances on the calendar.

**SELECTING AN EVENT INSTANCE**
Click to select any event instance on the calendar and it will be displayed with a blue border on the top and bottom. Note that all other instances of the same object on the calendar will change background color at the same time for easy reference.

**RESCHEDULING VIA DRAG-AND-DROP**
After you select a future job or task instance, you can easily move it to a different time slot by dragging-and-dropping the instance. After you move an instance, you will receive a confirmation dialog box. Either click **OK** to confirm the move or **Cancel** to return the instance to its original position.

**NOTE:** This feature is disabled in the evaluation version of SentryOne.

**VIEWING STEP DETAILS**
To view full step details for an event instance, click the small gray triangle in the lower-right corner of the instance and the step callout window will appear. You can move the window by clicking the gray bar at the top, and you can resize it by clicking the small dashed triangle in the lower-right corner. Note that event instances that don’t contain steps, such as status events or Windows tasks, will not have a step callout button.

8.3 List View

**LIST VIEW**
To view event history in a list format, select **List** from the **View** toolbar. The list view shows all event data that would otherwise be displayed on the calendar, but in a list format. Click the [+] icon to expand and view the full step details for the event, if any exist. Failed instances will be
highlighted in red.

**NOTE:** The history List View for one day can be opened by pressing (CTRL + ALT + H) when an applicable object is selected.

**SPLIT VIEW**

You can show both the Calendar View and the List View simultaneously by choosing Split from the View toolbar. When you select an event instance on the calendar, it will be auto-selected in the list window, and vice versa.

**GROUPING AND SORTING**

You can drag-and-drop any column heading into or out of the area directly above the list view to change the active grouping. You can also click any column heading to sort by that column.

8.4 Custom Event Views

Custom Event Views are used to tailor views to view and monitor specific activity across the enterprise.

For example, if you wanted to monitor all backup jobs across all your servers to determine if there is contention for a backup device, you could easily bring up that specific information to your calendar view and drag-and-drop those jobs to level your backup schedule or view performance information. After you set up a Custom Event View, it will operate like other views selected from the Navigator pane. Custom Event Views may also be set as the Default View when the SentryOne Client starts by right-clicking the view node and selecting the appropriate context menu item.

See Working with Custom Event Views for details on proper setup and configuration.

**NOTE:** To see new Custom Event Views created by other Client users, right-click on the main Custom Event Views node and select Refresh.

8.4.1 Working With Custom Event Views

From the Navigator pane expand “Custom Event Views”. This will provide the Sub-Nodes: “Local” and “Shared”. Local Views are only available to the User that created them. Shared Views are available to any User on any SentryOne Client. The setup of the View is the same for either option.

To add a new Custom Event View, double-click on the Local or Shared Node or select “New” from the context menu. This will bring up the “Add a Custom Event View” window in the workspace. The “Creator” field is already populated with your username and is uneditable.
Complete the “Name” and “Description” fields with information that will make sense to anyone who will use the view.

**ADD INSTANCES**

Next, click on the “Add Instances...” button to add objects that will be included in the view. The top box allows you to select by individual Instances, including Windows Task Schedulers. Additionally, it allows you to specify all Instances of a specific type, i.e. All SQL Server Instances. The middle box allows you to select entire SQL Server groups. Finally, the bottom section allows you to filter for specific objects from those Instances. This specifies which objects will be monitored by performance counters in this view.

**IMPORTANT:** Instance text filters directly control which event objects in the view will be considered monitorable when performance monitoring is enabled for a view, versus the General tab text filters which control only which objects will be displayed on the active Calendar/List View. Instance text filters are always applied first, before General text filters. It's very important to ensure the Instance-level text filters are set properly to avoid monitoring other objects unnecessarily. Please see Filtering Process topic for more details.

Click “OK” to save your Instances and return to the View Configuration window. Note the text box next to the “Add Instances ...” button verifies that Instances are now assigned to this View.

**SELECT OBJECTS**

Next, click on the “Select Individual Event Objects” button to select additional individual objects to view

The “Select Objects” window is broken down into three sections. Use the top section to search for objects by type, object name, or step text. Once the criteria are entered, click the “Search” button and the results will be populated in the middle “Search Results” section.

Select objects to add by clicking on the object in the “Search Results” section and click on the “Add” button.

The “Selected Objects” section show what objects will be included in the view.

Click on the “OK” button to save your selections and return to the View Configuration window.
Note that the text box next to the “Select Individual Event Objects” button verifies that objects are now assigned to this View.

**SELECT EVENT SOURCES**

Next, click on the "Select Event Sources" button. This is where you will define which Event Sources will be associated with the View. For example, if you only wanted SQL Agent Jobs in the View, ensure that the "Show SQL Server Agent Jobs" checkbox is checked and the remaining sources in the listbox are unchecked. You also have the option of selecting specific objects or categories when available.

**SET FILTERS**

Finally, click on the “Set Filters” button to select specific filters to apply to the View. The filter tabs will be restricted by the Instance types selected previously and the event sources available on those Instances. For example, if Reporting Services doesn’t exist on any selected Instances, the Reports tab will not be displayed.

*IMPORTANT*: General tab text filters are used only to control which event objects are displayed on the active calendar/list view. They do not affect which objects will be considered monitorable when performance monitoring is enabled for a view - this is directly controlled by the Instance text filters, which are always applied first, before the General text filters. Please see the [Instances](#) and [Filtering Process](#) topics for more details.

After configuring your filter settings, click “OK” to save and return to the View Configuration window. Note the text box next to the “Set Filters ...” button verifies that the filter is set for this View.

**SAVE YOUR VIEW**

Click on the Save button on the toolbar and it is ready to use.

Your new View will be added to the tree menu in the Navigator pane. Just double-click on it to open it, or expand the node to add Performance Counters ('Generating Performance Alerts' in the on-line documentation).

**SAVE VIEW AS... MENU ITEM**

Another way to create a Custom Event View is by using the Save View As item under the File menu. Anytime a particular calendar view is active that contains the Instances and filter settings you would like to save for a Custom Event View, simply go to File > Save View As. You will be asked for a name and optional description for the view. You may also check the box to make it public. This places it under the Shared node in Custom Event Views for use in any SentryOne Client.

*NOTE: If a view has performance monitoring enabled, using Save View As does not transfer performance counter settings to the new view.*

Creating a new Shared Custom Event View will trigger an Audit: Custom Event View: Created condition. Deleting and/or modifying existing Shared Custom Event views will also trigger similar Audit: Custom Event View: Deleted or Audit: Custom Event View: Modified conditions in SentryOne.
ORGANIZING CUSTOM EVENT VIEWS

Both Local and Shared Custom Event Views can be organized using custom folder names. Create a folder by selecting "New Folder" from the context menu on the Shared or Local nodes. Once folders are created, views can be moved between folders by dragging and dropping the view into the desired folder.

See also:
- Filtering Process
- Conditions and Actions

8.4.2 Filtering Process

When setting up a Custom Event View, it is important to understand the two levels of object filtering that take place. The first level of filtering, source level filtering, determines which objects are included in the view. Any performance counters that are applied to this view will be applied to every object that is included after this level of filtering. The second level of filtering, general filtering, only determines under which circumstances the above objects will be visible on the calendar. They will still have performance counters applied to them whether they are visible or not.

SOURCE LEVEL FILTERING

Any filter text specified from the Add Instances window when setting up or making changes to a view is at the source level. This will restrict the number of objects associated with the view and therefore the number of objects that will be monitorable.

After the view is created, additional source level filtering can be applied through the Event Sources tab such as Jobs, Alerts, Tasks, etc. By checking or un-checking the Show <source> checkboxes, you determine whether objects of this type will be included in the view.

Also on the Event Sources tab are Event Objects and Categories boxes that allow you to further filter objects from the view. These boxes are typically populated when working with a view with only one Instance.

GENERAL FILTERING

Any filters applied from the Filter tab of the Event Views pane only determine whether or not certain event objects will be visible on the calendar. Even if they are made invisible by this filter, they will still have any performance counters applied to them that are applied to the view. This is an important distinction for performance purposes. If you do not want a performance counter activated for an object in a view, you must ensure that it is filtered at the source level, and not simply made invisible by the Filter tab.

Please view the Event View Filter Process Flowchart for an illustration of this process.

8.4.3 Filter Process Flowchart

EVENT VIEW FILTER PROCESS FLOWCHART
8.4.4 Leveling Backups Across a SAN or NAS

Sample views can be easily modified to further restrict event instances shown to only those
utilizing the same "shared resource", such as a NAS (Network Attached Storage) or SAN (Storage Area Network) device, or even a network segment. In this example, we'll demonstrate the simple steps required to modify an existing sample view to create a "shared resource view" for all Red-Gate backups writing to a SAN, for the purpose of leveling the backup activity across the SAN to eliminate any bottlenecks, thus reducing contention and maximizing performance of the SAN and network.

From the Navigator pane expand “Custom Event Views.” This will expose the “Local” and “Shared” sub-nodes. Local views are only available to the user that created them. Shared views are available to any user on any SentryOne Client. The setup of a view is the same for either option.

If you expand the “Shared” node, you’ll notice the pre-existing “Sample Views” sub-node. Click the sample view for the backup system you use. (In this example we’ll use the sample view for Red-Gate SQL backups.)

The first time you double-click on a sample view or select “Open” from the context menu, the following message box will appear.

Clicking “Yes” will bring up the “Select Instances” box.

Note that the box has three sections, “Instances”, “SQL Server Groups”, and “Instance Text Filters”. This allows you to select individual Instances (SQL Server or Task Scheduler), all SQL Servers in a group, or a combination of both. Since we want to see all Red-Gate backup jobs across all SQL Servers in the enterprise, we’ll select "All" under SQL Server Groups.

The key to each sample view is the “Instance Text Filter.” With it you can easily restrict the jobs placed in the view by the actual TSQL or CmdExec text used in a job’s steps. Multiple keywords are separated with a semicolon, and a plus sign (+) is used in front of any keyword(s) that you want to apply using an "and"...otherwise the keyword will be applied with an "or".

Note the default "Step Text” filter for the Red-Gate SQL backups sample view: "master..sqlbackup; SqlBackupC”. This filter will return only jobs which contain any of those strings somewhere in the step command text. It will return all matching backup jobs, not just those writing to a particular SAN.

To restrict to only those jobs writing to a SAN, one minor adjustment to the filter should be made, as shown below. Add an "and" filter using the base SAN name or IP address. In this example we’ve used "+\SAN01". This will cause the view to return any jobs which contain "master..sqlbackup", “SqlBackupC” AND contain \SAN01.

After clicking “OK”, we have a Custom Event View ready to use.
The jobs with an orange background indicate scheduling conflicts - note we have several cases where backups coming from different servers are hitting the SAN at the same time. Drag-and-drop functionality can be used on future job instances on the calendar to eliminate the scheduling conflicts and reduce associated contention on the SAN or network.

When all jobs have a white background, you know you have a “leveled” schedule. A leveled schedule ensures that any bottlenecks and associated IO errors on the SAN's disk subsystem and network are minimized, and that backup jobs run at optimal speed!

See Also:

Save View As... Menu Item

8.5 Chaining and Queuing

8.5.1 Event Chains

An event chain is a series of dependencies between events running on the same server or different servers. There is no limit to the length of a chain, or the number of events that can be defined for a chain.

To navigate to event chains, select the “Event Chains” node from the Navigator.

Name - Contains the name of the Event Chain. This also is the value displayed in the treeview under the Event Chains node.

Description - Descriptive text for the event chain.

Created by - Displays the chain creator and the date/time of chain creation.

Updated by - If the chain is updated, displays the updater and date/time of update.
Select Chain Instance - Use this drop-down list to select a past runtime for the chain to view its history.

End Time - When viewing an historical chain instance, this will show the time the chain completed.

Status - Displays the status of the chain being viewed as, “Ready”, “Running”, or "Completed".

Chain Enabled - Option to enable or disable the event chain.

Edit Button - To add nodes or make changes to a chain, click this button.

Auto refresh - Determines how often the chain workspace will refresh to show updated information when the chain is currently running.

Note: Only the initial job of the Event Chain needs to be scheduled for the chain to run.

EVENT CHAIN EXAMPLES

Example A: When Job1 (SQL Server job) completes, then execute Job2 (SQL Server job). If Job2 succeeds, execute Task A (Windows Task Scheduler Task). If Job2 fails, execute Job3
In order to accommodate a path that encompasses the same event more than once and avoid a circular reference, the concept of nodes are used. A Node is simply a unique representation of an event. Even though Job1 is listed twice, the chain can treat the event as two separate instances and apply varying conditions to each instance which provides a greater degree of flexibility.

**CIRCULAR REFERENCES**

An event chain is a logical workflow through a series of one or more event objects, with each occurrence of an event object represented by a unique event node. A circular reference would occur if the workflow was built to reference an event node a second time. Such a workflow is not allowed in an event chain.

**For example:** When Event A completes, then execute Event C. When Event C completes, then execute
To allow an event to occur a second time in an event chain, add the object Event A a second time to the object list. Change the To Node in the second link to reference the newly created node. This method can be repeated to allow the event to occur a discrete number of times.

ADD AN EVENT CHAIN

- Double click the the Event Chains node in the Navigator pane or use the New context menu command.
- Assign the event chain a name and optionally a description.
- Right-click to add objects that will be used in the event chain or drag and drop jobs from the Navigator onto the Event Chain diagram. Node names will be given to each object.
- Add links to the chain by right-clicking the appropriate node, then select Workflow. Next, select the status condition for execution and the target node.
- Nodes can be drag and dropped to reposition for easier viewing of the chain. The Auto Layout toolbar button may also be used once links are established.
- Save the chain.

EDIT AN EVENT CHAIN

- Select Event Chain node from the treeview in the Navigator window.
- Click the Edit button.
- Edit the necessary properties of the Event Chain.
- Save the chain.

REMAPPING NODES IN AN EVENT CHAIN

Event Chain nodes can be remapped to other event objects. This can save you time over deleting and recreating nodes in your Event Chain. When you remap a node in an Event Chain the next step(s) in the workflow will remain intact. To remap a node use the Remap Node command available through the node context menu.
**Important:** If any Event Chain nodes or workflow steps are changed, including the remapping of a node, the history for the chain becomes invalid and will be lost.

**CHAIN HISTORY**

To view an historical run of a chain, select the runtime from the **Select Chain Instance** listbox. Runtime and status information will be displayed for each node. If the object completed successfully, it will have a green background. If it failed, it will have a red background. This is also shown for currently executing chains. Currently executing nodes will have an orange background.

**CONTEXT MENUS**

**NAVIGATOR PANES**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Opens the Event Chain.</td>
</tr>
<tr>
<td>Edit</td>
<td>Opens the Event Chain in edit-mode.</td>
</tr>
<tr>
<td>Start Chain</td>
<td>Starts all lead nodes of this Event Chain.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the chain.</td>
</tr>
<tr>
<td>Disable</td>
<td>Disable the chain.</td>
</tr>
<tr>
<td>Reset</td>
<td>Reset a currently running chain.</td>
</tr>
<tr>
<td>Change Folder</td>
<td>Allows you to move the specified chain to a different organizational folder.</td>
</tr>
</tbody>
</table>

**VIEW MODE**

**Workspace**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh</td>
<td>Refresh the current view. Also done by pressing <strong>F5</strong>.</td>
</tr>
<tr>
<td>Print</td>
<td>Print the chain diagram.</td>
</tr>
<tr>
<td>Print Preview</td>
<td>Preview the printed chain layout.</td>
</tr>
<tr>
<td>Zoom</td>
<td>Zoom in or out to view entire chain.</td>
</tr>
</tbody>
</table>

**Node - <Current> Chain Instance**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Node</td>
<td>Allows you to manually run a node in a chain. If the node is a lead node for a chain, then any child nodes will be executed upon completion. No child nodes will start if the node being executed is not a lead node.</td>
</tr>
</tbody>
</table>
Jump To ...

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- This Instance</td>
<td>Brings up a calendar displaying the selected node's current instance.</td>
</tr>
<tr>
<td>-- Last Failure</td>
<td>Brings up a calendar displaying the last failure for the selected node.</td>
</tr>
<tr>
<td>-- Last Instance</td>
<td>Brings up a calendar displaying the last instance of the selected node.</td>
</tr>
<tr>
<td>-- Next Instance</td>
<td>Brings up a calendar displaying the next instance of the selected node.</td>
</tr>
<tr>
<td>-- Navigator</td>
<td>Navigates to the selected node in the Navigator.</td>
</tr>
</tbody>
</table>

Show

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- History</td>
<td>Displays the last 4 hours of job history for the selected node.</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>---- 4 Hours</td>
<td>Displays the last 1 day of job history for the selected node.</td>
</tr>
<tr>
<td>---- 1 Day</td>
<td>Displays the last 3 days of job history for the selected node.</td>
</tr>
<tr>
<td>---- 3 Days</td>
<td>Displays the last 7 days of job history for the selected node.</td>
</tr>
<tr>
<td>---- 7 Days</td>
<td>Displays the last 14 days of job history for the selected node.</td>
</tr>
<tr>
<td>---- 14 Days</td>
<td>Displays the last 30 days of job history for the selected node.</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show</td>
<td>View the properties of the object.</td>
</tr>
</tbody>
</table>

Node - Historical Chain Instance

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Node</td>
<td>Allows you to manually run a node in a chain. If the node is a lead node for a chain, then any child nodes will be executed upon completion. No child nodes will start if the node being executed is not a lead node.</td>
</tr>
</tbody>
</table>

Jump To ...

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- This Instance</td>
<td>Brings up a calendar displaying the selected node's current instance.</td>
</tr>
<tr>
<td>-- Last Failure</td>
<td>Brings up a calendar displaying the last failure for the selected node.</td>
</tr>
<tr>
<td>-- Last Instance</td>
<td>Brings up a calendar displaying the last instance of the selected node.</td>
</tr>
<tr>
<td>-- Next Instance</td>
<td>Brings up a calendar displaying the next instance of the selected node.</td>
</tr>
<tr>
<td>-- Navigator</td>
<td>Navigates to the selected node in the Navigator.</td>
</tr>
<tr>
<td>Show</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>---- History</td>
<td></td>
</tr>
<tr>
<td>---- 4 Hours</td>
<td>Displays the last 4 hours of job history for the selected node.</td>
</tr>
<tr>
<td>---- 1 Day</td>
<td>Displays the last 1 day of job history for the selected node.</td>
</tr>
<tr>
<td>---- 3 Days</td>
<td>Displays the last 3 days of job history for the selected node.</td>
</tr>
<tr>
<td>---- 7 Days</td>
<td>Displays the last 7 days of job history for the selected node.</td>
</tr>
<tr>
<td>---- 14 Days</td>
<td>Displays the last 14 days of job history for the selected node.</td>
</tr>
<tr>
<td>---- 30 Days</td>
<td>Displays the last 30 days of job history for the selected node.</td>
</tr>
<tr>
<td>-- Runtime stats</td>
<td>Shows the current runtime stats for the selected node.</td>
</tr>
<tr>
<td>Start Chain Here</td>
<td>Allows you to execute a chain from a node other than a lead node. When this option is selected, a new chain instance is created that contains a copy of the nodes status for the instance you are viewing.</td>
</tr>
<tr>
<td>Properties</td>
<td>View the properties of the object.</td>
</tr>
</tbody>
</table>

**EDIT MODE**

**Workspace**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Nodes</td>
<td>Allows you to add new Jobs or Tasks as nodes in the chain.</td>
</tr>
<tr>
<td>Save</td>
<td>Save any changes to the chain.</td>
</tr>
<tr>
<td>Print</td>
<td>Print the chain diagram.</td>
</tr>
<tr>
<td>Print Preview</td>
<td>Preview the printed chain layout.</td>
</tr>
<tr>
<td>Auto Layout</td>
<td>Organizes and arranges the chain into a default flowchart pattern.</td>
</tr>
<tr>
<td>Zoom</td>
<td>Zoom in or out to view the entire chain.</td>
</tr>
</tbody>
</table>

**Node**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow</td>
<td>Allows you to specify the condition on which one node will execute another.</td>
</tr>
<tr>
<td>Rename Node</td>
<td>Rename the node. This will <strong>not</strong> rename the actual job or task.</td>
</tr>
<tr>
<td>Delete Node</td>
<td>Delete the existing node from the chain.</td>
</tr>
<tr>
<td>Properties</td>
<td>View the properties of the job or task.</td>
</tr>
</tbody>
</table>
### EVENT CHAIN NOTIFICATIONS

#### GENERAL ACTIONS

<table>
<thead>
<tr>
<th>Condition</th>
<th>Fired When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Chain: Started</td>
<td>Any lead node for an Event Chain has completed.</td>
</tr>
<tr>
<td>Event Chain: Completed</td>
<td>After last node in an Event Chain has completed.</td>
</tr>
<tr>
<td>Event Chain: Success</td>
<td>Every node in an Event Chain runs.</td>
</tr>
<tr>
<td>Event Chain: Failure</td>
<td>One or more nodes in an Event Chain do not run.</td>
</tr>
<tr>
<td>Event Chain: Runtime Threshold Max</td>
<td>The Event Chain runtime exceeds the Maximum Runtime Threshold settings specified for the Event Chain.</td>
</tr>
<tr>
<td>Event Chain: Runtime Threshold Min</td>
<td>The Event Chain runtime exceeds the Minimum Runtime Threshold settings specified for the Event Chain.</td>
</tr>
<tr>
<td>Event Chain Node: Started</td>
<td>The Event Chain node starts.</td>
</tr>
<tr>
<td>Event Chain Node: Completed</td>
<td>The Event Chain node completes.</td>
</tr>
<tr>
<td>Event Chain Node: Success</td>
<td>The Event Chain node completes, reporting a success.</td>
</tr>
<tr>
<td>Event Chain Node: Failure</td>
<td>The Event Chain node completes, reporting a failure.</td>
</tr>
<tr>
<td>Event Chain Node: Runtime Threshold Max</td>
<td>The Event Chain node exceeds the Maximum Runtime threshold specified for the node.</td>
</tr>
<tr>
<td>Event Chain Node: Runtime Threshold Min</td>
<td>The Event Chain node exceeds the Minimum Runtime threshold specified for the node.</td>
</tr>
</tbody>
</table>

When more than one workflow path exists in a chain, the Chain Success and Chain Failure conditions become less valuable since they are based exclusively on whether or not all nodes completed. At that point the Node Success and Node Failure conditions should be used for more accurate status information about the respective nodes and/or paths.

### SETTINGS

Event Chains and Event Chain Nodes support Runtime Threshold settings. These settings are specified as absolute runtimes for both the minimum and maximum runtime thresholds.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit From Parent</td>
<td>Specified whether the runtime threshold settings are inherited from the parent object.</td>
</tr>
<tr>
<td>Maximum Runtime Threshold</td>
<td>The maximum allowed runtime for the chain/node.</td>
</tr>
<tr>
<td>Minimum Runtime Threshold</td>
<td>The minimum allowed runtime for the chain/node.</td>
</tr>
</tbody>
</table>
AUDIT ACTIONS

<table>
<thead>
<tr>
<th>Condition</th>
<th>Fired When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Chain Modified</td>
<td>Modifications are made and saved to an Event Chain.</td>
</tr>
</tbody>
</table>

EVENT VIEW

All Event Chains contain an Event View sub-node. Double-clicking or selecting Open from the context menu will produce a calendar of all the objects that are apart of the selected chain. As with any Custom Event View, you also have the ability to add Performance Counters to any Event Chain View.

EVENT CHAIN CALENDAR INFO

All Event Source Objects that appear on a calendar because of an Event Chain will display a small chain glyph in the top left corner.

Additionally, the Event Source Object tool tip will display information about any chain(s) that this object is associated with. An * is placed next to the actual chain that created this instance.
REPORTING SERVICES JOBS IN EVENT CHAINS

It is a known limitation that Reporting Services jobs will not work within chains in versions prior to 2.7. This limitation occurs because Reporting Services jobs are filtered out of SentryOne as they have no duration and do not reflect the accurate information about the report.

A workaround is available in version 2.7 and later, as described below:

Normally, SentryOne filters out a Reporting Services job. You can force them to stay in the history and thus alert chaining that they have completed by placing an additional space after the AddEvent call.

For example, if you currently have:

```
AddEvent @EventType='Test', @EventData='5B0537EA-7A87-4C8B-B002-03A51DD5100C'
```

You could change it to:

```
AddEvent  @EventType='Test', @EventData='5B0537EA-7A87-4C8B-B002-03A51DD5100C'
```

After the change is made, reset the chain.

**IMPORTANT**

If you manually modify the SSRS scheduler job step text as described above, and at a later date make modifications to the report schedule via any SSRS tool, the manual step text changes you made will be overwritten. It is important to remember to make these modifications to the step text again if you have included an SSRS report as node in an Event Chain.

COPY CHAIN

You may run into a situation where you want to create a second chain that is similar to an existing chain, or want to use an existing chain as a template for new chain. You can effectively copy a chain, using the Save Chain As... option on the File Menu,

You will then be asked to enter a new name and optional description for the chain.

**NOTE:** To see new Event Chains created by other Client users, right-click on the Event Chains root node and select Refresh.

ORGANIZING EVENT CHAINS

Event Chains can be organized using custom folder names. Any Event Chain can be moved into another folder by dragging and dropping into another folder.

See also:

- Event History Monitor Settings
- Custom Event Views
- Performance Monitoring (`Schedule Performance Monitoring` in the on-line documentation)
- Conditions and Actions

8.5.2 Job Queueing

Job Queueing allows a resource intensive job to delay other jobs from executing until it has completed. This allows the resource intensive job to utilize more of the server resources, allowing all jobs to finish faster and more efficiently. Unlike Event Chains, Job Queueing applies only to the jobs on an individual SQL Server instance.
Each SQL Agent job has its own queuing settings. Simply click on the desired job and select the "Settings" tab from the Conditions and Settings Pane. Note that you can also configure default queuing settings at both the Global and Instance levels, and those settings will be inherited by any associated jobs.

BEHAVIOR WHEN QUEUING OTHER JOBS

INHERIT FROM PARENT
Select False if you want to configure the queuing behavior differently than the inherited settings.

QUEUE TYPE
Under Queue Type there are 3 options:
1. Never queue – This job will not queue any other jobs.
2. Queue for specified time – This job will queue other jobs for the specified time.
3. Queue indefinitely – This job will queue other jobs until this job has completed.

QUEUE OTHERS FOR UP TO
This is the maximum amount of time this job will queue other jobs. This setting is not available Globally, since it could be disastrous if it were accidentally enabled for all jobs on all SQL Servers.

AUTO-START THRESHOLD
Any queued job whose next scheduled run time is beyond the specified threshold will be started automatically upon leaving the queue. If a job's next scheduled run time is before the threshold it will not auto-start, but resume with its next scheduled run. This setting is extremely valuable for avoiding having to configure the auto-start type for every possible job that can be queued. It
effectively provides a safeguard so that jobs that don’t run very frequently will auto-start and thus not miss a scheduled run, and those that do run frequently will resume their schedules since it usually doesn’t matter if they miss a few runs. It also helps to automatically even the load when a queuing job finishes, so that all queued jobs don’t auto-start at the same time and cause resource contention issues.

Note: The maximum number of jobs that can be queued on a server is specified by the Maximum Queue Length setting under SQL Server Instance settings.

BEHAVIOR WHEN THIS JOB IS QUEUED

INHERIT FROM PARENT
Select False if you want to configure the queuing behavior differently than the inherited settings.

QUEUE TYPE
Under Queue Type there are 3 options:

1. Never queue – This job cannot be queued by another job.
2. Queue for specified time – This job will only be queued for the specified time.
3. Queue indefinitely – This job can be queued indefinitely.

QUEUE FOR UP TO
This is the maximum amount of time this job can be queued.

AUTO-START TYPE
Controls the auto-start behavior for the job when it leaves the queue.

There are 3 Auto-Start options:

1. Don’t Auto-Start, resume schedule - Skip the job until it’s next scheduled run.
2. Auto-start immediately - If this option is checked the job will execute as soon as it leaves the queue. However, sometimes you may not want this behavior, such as for jobs that should only run at a specific time, or for recurring jobs (jobs that run multiple times per day) where a single missed run is not critical.

   NOTE: If this option is not selected and the job is always queued by another job, such as two jobs with the exact same schedule, it may never run.

3. Use default setting - The job will use the queuing job's "Auto-Start Threshold" setting. This is the default setting, and is recommended for most cases.

   Note: Jobs which use shared schedules on SQL Server 2005 and above will not be queued.

QUEUE LOG

The Queue Log can be viewed by selecting View Queue Log from the context menu of the Jobs node of a SQL Server or an individual job node. The Queue Log provides details about all recent queuing activity that has taken place on the server or for the particular job.
AN EXAMPLE SCENARIO

The primary reason for queuing jobs is to ensure a high priority, resource intensive job has full access to the server's resources to complete its task without competing with other jobs that can be delayed until the high priority job is done.

For example, consider a full backup job that runs once a week, every Sunday at 1 AM. If this is the only job running, having full use of the server's resources, it can complete in about 45 minutes. However, there are typically other recurring jobs on the same server that can cause contention for resources, resulting in the backup taking almost two hours to complete. This in turn causes delays in many other recurring jobs being able to successfully complete. While it is not plausible to reschedule a five minute recurring job to recur once an hour for the sake of this one weekly backup, it may be acceptable to temporarily queue that job for one hour once a week until the backup is complete.

8.6 Clearing Failed Instances

IDENTIFYING A FAILED INSTANCE

When a certain event has failed, there are several ways the SentryOne Client signals you to that fact.

In the Navigator Pane, the text of the object will be red to indicate a failed instance. In the Calendar View, the status bar for that event's instance will be red.

and in the List View, the first cell has a red background.
CLEARING THE FAILURE

In order to clear a failure, right-click on any of the above mentioned instance views. Select **Clear Failed Status**. You will be asked if you want to leave a note before clearing.

Alternately, select **Add/Edit Notes** from the context menu and the Notes window will appear. Enter details about the failure and resolution taken, then click **Save**. A message box will appear asking if you would like to clear the failed status for the object. Click **Yes**, and all open failures for the object will be cleared, and **Navigator** nodes shown in red will be changed back to black. When clearing the failed status for a maintenance plan job, maintenance plan, etc., any related child or parent objects will also be cleared.

See also:

- Notes
- Reporting

### 8.7 Copy Job/Alert Wizard

SentryOne allows you to copy any job or alert to another server or servers quickly and easily with the Copy Job/Alert Wizard. This can be done by:

- Right-clicking a job or alert instance on the SentryOne Client and selecting the **Copy Job To..** or **Copy Alert To..** from the context menu.
- Next, select the servers to which you would like to copy the object.
- Click **Next**, confirm the target servers and click execute. The objects will be copied and readily available on the target servers.
- Click **Finish** to close the wizard.

### 8.8 Conflict Viewer
Event conflicts are often one of the greatest contributors to performance problems on a SQL Server. The **Conflict Viewer** provides valuable insight into how event schedules are contending with each other, and enables quick and easy cleanup of conflict conditions, ensuring your event schedules are leveled. To open the Conflict Viewer, click the **View Conflicts** button on the Toolbars.

The view is initially sorted descending by “Total Conflict Time”, which is the total time an event conflicts with other events. This measure is extremely valuable in determining which event objects may be causing the most resource contention. The primary event object is listed with its detail information including Name, Start Time, End Time and Duration. The event objects that are conflicting with it are listed below it along with their detail information, as well as the actual time in conflict.

**Double-click any conflict record to jump directly to the corresponding job instance on the calendar.** From there you can use drag-and-drop to reschedule the associated events and clean up the conflict condition. This is the quickest and most effective way to level an event schedule.

**NOTE:** The Drag-and-Drop feature is disabled in the Evaluation copy of SentryOne.

### VIEWER SETTINGS

**Show Conflicts Over**

Use this setting to set the time range over which you want to view conflicts. The Conflict Viewer will start at the current time and project out into the future by the specified amount

**Minimum Time in Conflict**

Use this setting to eliminate noise caused by short-running events.

---

8.9 **Event Object Information Window**

When you right-click any **event object** in the Navigator and select Open or double-click an **event instance** on the Calendar, Navigator, or **Runtime Graph**, a tab for the event object's Information Window will appear. The window contains various metadata about the object. The information displayed will change based upon the type of object.
For active event objects, runtime statistics and the Calculate Runtime Statistics Since option will also be shown, as well as schedules for schedulable objects such as jobs, tasks and reports. Tasks include additional data such as the option to define a success exit code.

**RECALCULATING RUNTIME STATISTICS**

On the information window for all active event objects you'll see the Calculate Runtime Statistics Since option. Use this feature whenever a major change is made to an event definition which may cause it's typical runtime to increase or decrease dramatically from what it was previously, to avoid erroneous runtime minimum or maximum notifications from being sent. Simply check the box next to Calculate Runtime Statistics Since and enter the date and time the event was changed, then click Save on the toolbar. This will cause SentryOne to immediately recalculate runtime statistics using only run history since the specified date/time. This value will also be used whenever the SentryOne Monitoring Service auto-calculates statistics for the event in the future.

---

8.10 Multi-server Administration (MSX/TSX)

SentryOne provides additional functionality for multi-server administration. Any server designated as a master server will have "(MSX)" after its name in the Navigator pane. You will also see an additional sub-node called "Multi-server Jobs". This node provides the same functionality and context menu items as the standard Jobs node, however, jobs under this node are distributed to target servers. Just like the "Jobs" node, if you open it (by single- or double-clicking) you will see a calendar/list view of all multi-server jobs across all target servers - by default this view is filtered to show only failed or long running historical events, just like the global and group views. The filter criteria can be changed in the Filter pane.

Target servers appear in the Navigator pane with "(TSX: <SERVERNAME>)", where SERVERNAME is the name of the master server to which it is enlisted.

**PUBLISHING CONFIGURATION TO TARGET SERVERS**

Just like standard local jobs, Conditions and Actions can be explicitly configured for multi-server jobs by single clicking a multi-server job node in the Navigator pane. Also, just like local jobs, multi-server jobs on a master server inherit any Conditions explicitly configured for the master server connection (accessed by single-clicking the master server node). Bear in mind, however, that multi-server jobs don't actually execute on the master server - they are created and configured on the master server and then pushed out to one or more target servers for execution.

In other words, the multi-server jobs exist on the master server strictly for configuration purposes,
and this applies to **watched** status and actions as well. These settings for multi-server jobs don’t actually do anything on the master server - they exist only so that they can easily be applied to target servers, to avoid having to manually change configurations on each target server individually. This is accomplished by the **Publish Configuration Wizard**. When a change is made to configured Conditions or watched status for multi-server jobs on the master server, those changes can easily be pushed out to selected target servers using the wizard.

The Publish Configuration Wizard can be run for an individual multi-server job, or at the master server level. To start the wizard, right click any job under the “Multi-server Jobs” node, or at the Instance level by right clicking on the MSX server node.

Select **Publish Config to Targets**, and the **Publish Configuration Wizard** appears. After clicking **Next** at the introduction, select the target servers to which you wish to publish conditions and watched status. If you launch the wizard from the master server node, you have the option of publishing watched/unwatched settings, SQL Server Instance-level conditions, and/or job-level conditions. If you launch from an individual job, you may only publish the watched setting and conditions for that job; the Instance-level conditions option is unavailable.

After making your selections, the wizard will confirm your actions and execute the script to make the changes to the specified target servers. All specified targets are now configured identically to the master server, based on the options you selected.

**NOTE:**

- **Target (TSX) SQL Servers are really no different than any other SQL Servers managed by SentryOne.** Other than the fact that they can include a mix of local and multi-server jobs, the hierarchical configuration process used by SentryOne applies to target servers and all jobs on target servers just as with any other Instance or object. For example, if you never run the Publish Actions Wizard, but have configured global actions, the global actions will be inherited by all target servers as usual.

- **Likewise, if you have set Auto-watch new Objects to True for SQL Agent Jobs at the global level (either manually or using the Quick Start Setup Wizard), any local or multi-server jobs on a target server will be watched by default.**

- **There isn’t a two-way relationship for watched status and actions configuration between target servers and a master server.** Once you have published these settings to a target server from the master, you can easily override them later on the target server as usual, but watched status and actions changes made on a target server have no effect on the master server configuration.

8.11 Runtime Stats

The **Runtime Stats** node displays an active event object’s historical runtime information. Two views are provided: **Runtime History** and **Runtime Aggregates**. This selection can be made from the main toolbar. These graphs make it easy to identify runtime trends, which can be an indicator of or eventually lead to performance issues with the event or the server.

**RUNTIME HISTORY GRAPH**

This graph shows a job’s actual runtime and status over the specified interval. A red bar indicates that the instance failed.
RUNTIME AGGREGATES GRAPH

This graphs show the distribution of the job’s actual runtime over the specified interval. This data can be valuable when deciding where to place a job in the schedule to minimize contention.

GRAPH CONTROLS

Use the Navigation toolbar to change the start date/time and interval for the graph. The Days control can go up to 60 days. Whenever the selected interval is less than 1 day (1 minute through 4 hours), the Days control will be changed to 1 day.

<table>
<thead>
<tr>
<th>Min/Max Y Val</th>
<th>When a runtime graph is first displayed, SentryOne will try to determine the most appropriate interval and maximum Y-axis value. You can change either of these settings at any time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-scale to Max Value</td>
<td>Checking this box will automatically select the appropriate Max Y Value of the data being shown in the current view.</td>
</tr>
<tr>
<td>Interval Type</td>
<td>Changes the interval used for the Y-axis.</td>
</tr>
<tr>
<td>Show Value Labels</td>
<td>Deselect this checkbox to hide the bar value labels. This can become necessary for broader ranges with hundreds or thousands of bars to reduce the noise on the graph.</td>
</tr>
</tbody>
</table>

For information on the Top SQL Runtime Stats, see Top SQL Runtime Stats.

8.12 Toolbars

SAVE AND PRINT TOOLBAR

The following describes the Save and Print Toolbar buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Save" /></td>
<td>Universal save button.</td>
</tr>
<tr>
<td><img src="image" alt="Print Preview" /></td>
<td>Displays a print preview window for the selected object.</td>
</tr>
<tr>
<td><img src="image" alt="Print" /></td>
<td>Prints the selected object.</td>
</tr>
</tbody>
</table>

NAVIGATION TOOLBAR
The Navigation Toolbar is used for navigating throughout the Calendar View as well as Performance Graphs and Runtime Stats.

**DATE/TIME SELECTOR**

The Date/Time Selector controls allow you to jump to a specific date or time, and select the desired time interval.

<table>
<thead>
<tr>
<th>Date Selector</th>
<th>Used to select the date to view on the calendar.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Selector</td>
<td>Used to select the time to view on the calendar.</td>
</tr>
<tr>
<td>Go</td>
<td>Moves the calendar to display the time selected in the date/time selector maintaining the currently selected interval.</td>
</tr>
<tr>
<td>Go to Current Time</td>
<td>Jumps directly to the current time maintaining the current selected time slice and interval, unless the interval is greater than 1 hour in which case 1 hour will be used.</td>
</tr>
</tbody>
</table>

When you click the Date Selector a small calendar will appear. Click any day to select it. Click the right and left arrows to change the month. Click the bold Today text at bottom to auto-select the current day. You can also type in values for the day or year directly.

To navigate to a specific time on the selected day, select the Time Selector by clicking a time increment (hour, minute, etc.) and then using the up/down arrows or typing in a new value.

The workspace view will not refresh until you have completed your selections and clicked the **Go** button.

**INTERVAL SELECTOR**

The Interval Selector is used to zoom the current time interval in or out.

| 1 Minute        | Displays the calendar in a one minute interval. |
| 10 Minutes      | Displays the calendar in a ten minute interval. |
| 1 Hour          | Displays the calendar in a one hour interval. |
| 4 Hours         | Displays the calendar in a four hour interval. |
| 1 Day           | Displays the calendar in a one day interval. |

**MISCELLANEOUS CONTROLS**

Sets the total range of days to be displayed on the calendar or graph. Click either arrow to decrease or increase the value,
| Days Selector | or simply type in a new value. The maximum value is 30 days. The currently selected interval (1 minute to 4 hours) as well as the start time of the range will be maintained when changing the number of days |
| Previous Interval | Used to navigate backward in time to the previous interval. |
| Previous Small Interval | Used to navigate backward in time using a smaller interval. |
| Next Interval | Used to navigate forward in time to the next interval. |
| Next Small Interval | Used to navigate forward in time using a smaller. |
| Refresh | Used to refresh the active workspace view by retrieving the latest information from the SentryOne database. Hold down the Ctrl key when you click it to force a hard refresh. This will cause the SentryOne Client to connect directly to and collect event data from the target Instance in case the Monitoring Service is down. (Not available on Custom Event Views or views with multiple Instances) |
| View Conflicts | Displays all schedule conflicts for the selected Instance. (See Conflict Viewer) |

**VIEW TOOLBAR**
Select from Calendar, List, or Split views. The active range will note change, only the format in which the event data is displayed. (See Calendar View)

8.13 Extended Job Step Logging / SSIS Logging

SentryOne Event Calendar is capable of collecting and displaying extended step log data for SQL Server Agent job steps on SQL Server 2005 and above. In order for this data to be collected, certain options must be enabled at the job step level.

These options are only available for the following step subsystems:
- Operating System (CmdExec)
- Analysis Services Command
- Analysis Services Query
- Transact-SQL script (T-SQL)

Refer to these topics for additional information:
- Extended Job Step Logging (SQL Server 2005+)
- Enabling SSIS Logging
### 8.13.1 Extended Job Step Logging (SQL Server 2005+)

By default the step history for SQL Server Agent jobs includes a maximum of 1024 characters of output, and it is not typically cleanly formatted. If you need to see more of the history data using the native output format, in SQL Server 2005 and above you can easily enable extended logging for a job step. Once enabled, SentryOne will collect and process the first 512KB of step output.

**Note:** Logging for SSIS (SQL Server Integration Services) packages should always be enabled using the SentryOne Client. See the Enabling SSIS Logging topic for more details.

To enable extended logging for a job step, open the properties for the job step in SQL Server Management Studio. Select “Advanced” on the left side of the form. Check “Log to table” to enable extended step logging, then click OK.

**Important:** Do not select the “Append output to existing entry in table” option, or SentryOne will not collect the extended step log data.

Once enabled, extended job step log data can be viewed in the SentryOne Client, both in job step callout windows on the calendar, and in history list views. A new level will be added underneath any job steps with extended logging enabled. Simply click [+] to expand the level and view the history details.

For job steps where SSIS Logging has been enabled through the SentryOne Client, the step log is parsed and displayed as one row per SSIS event.

### 8.13.2 Enabling SSIS Logging

On watched SQL Server 2005 and above instances, SQL Agent jobs which have one or more steps with a subsystem type of SQL Server Integration Services Package will have an SSIS Logging Options command within their context menu (this includes SQL Server Maintenance Plan jobs). The SSIS Logging Options command makes it easy to enable or disable SSIS Logging.

Once SSIS Logging has been enabled for a SQL Agent job, a new message text level will be added beneath any SSIS job steps, in both the Event Calendar step callouts, and in history list views. A row will be added for each SSIS event, up to the first 512KB of log data.

When you enable SSIS Logging for a SQL Agent job the following will take place:

1. All SSIS subsystem steps will be changed to CmdExec subsystem steps.
2. The append /CONSOLELOG and other appropriate parameters will be added to the command line of each SSIS step.
3. The Log to Table option will be set for each SSIS step.

See Extended Job Step Logging for more details.

**LOGGING LEVEL**

The SSIS Logging Options dialog allows you to set the Logging Level to one of the following.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>SSIS Log data will not be collected or displayed. SSIS Log data will be collected and displayed for the following package</td>
</tr>
</tbody>
</table>
Enabled

This is the recommended setting.

Enabled (Verbose)

Enabled

SSIS Log data will be collected and displayed for all package events. Since large amounts of data will be collected and processed with verbose logging, to avoid possible performance issues it is highly recommended that this setting only be used for troubleshooting purposes, and immediately set back to "Enabled" when complete.

Note: SentryOne will collect and process only the first 512KB of SSIS log data for each job step. Typically the only time a job would ever exceed this limit is when verbose logging has been enabled.

RUN AS

When enabling SSIS Logging, using the Run As drop-down list, you may select to run the package as follows:

Run As Description

Keep current proxy settings

When you select the Keep current proxy settings option, the account specified within the Job Step properties will be used to run the SSIS job step.

Note: The current proxy must have active rights to the Operating system (CmdExec) subsystem in order to enable SSIS logging within SentryOne.

SQL Agent Service Account

When you select the SQL Agent Service Account option, the SQL Server Agent Service Account will be used to run each SSIS job step, instead of the account specified within the Job Step properties.

Additional proxy accounts

Additional proxy accounts with the necessary Operating system (CmdExec) subsystem rights will also be listed as an option.

Changing the SSIS Logging options for a job will trigger the Event Object: Properties Modified Audit Condition within SentryOne.
SENTRYONE PERFORMANCE ANALYSIS

- **Dashboard**
  - Dashboard Resolutions/Retention
  - Baseline
  - Performance Metrics
- **Processes**
- **Disk Activity**
- **Disk Space**
- **Top SQL**
  - Plan Explorer (integrated)
    - Plan Explorer Sessions
    - SQLPerformance.com Plan Upload
    - Plan Diagram Options
    - Index Analysis
  - Top SQL Runtime Stats
- **AlwaysOn**
- **Query Plans**
- **Indexes tab**
  - Fragmentation Manager
- **SSAS Usage Totals**
- **Blocking SQL**
- **Deadlocks**
- **Quick Traces**
- **Performance Analysis Overview**
- **Performance Analysis Toolbar**
- **Data Grids**
- **Virtual Machine Support**
- **cloud.sentryone.com**
  - Server Details
  - Server Health Status

OPENING THE PERFORMANCE ANALYSIS DASHBOARD

The Performance Analysis Dashboard can be opened by selecting (Open > Performance Analysis) from the context menu of any Instance. It can also be opened by double-clicking the Instance and choosing Performance Analysis from the product selection form.

PERFORMANCE ANALYSIS TABS

When using SentryOne to monitor SQL based targets and instances (SQL Server, SSAS or Azure...
SQL Database) the interface consists of the several tabs. Not all tabs are included for all target and instance types.

The Dashboard tab contains relevant performance data for both the physical server and SQL Server, all on one screen. Use it to view real time data or historical data for any point in time. The Performance Metrics section of the User Guide contains a detailed description of the various performance metrics displayed on the Dashboard.

The Disk Activity tab contains a graphical view of your server's disk topology. Use the Disk Activity tab to quickly pinpoint performance bottlenecks at the database file, physical disk, or controller level for SAS, NAS, or direct attached storage.

This tab is not displayed for an Azure SQL Database target.

The Disk Space tab contains information about disk space utilization within your environment by database and file. Use the Disk Space tab to easily identify disk capacity issues and understand where various database files reside on the disk system.

This tab is not displayed for an Azure SQL Database target.

The Indexes tab displays information that has been collected about your tables and indexes by Fragmentation Manager. The Indexes tab can be used to help you make intelligent decisions about index management in your environment: such as when and how to perform defragmentation operations, when to adjust fill factors, or even when an index definition should be changed.

This tab is not displayed for an Azure SQL Database target or SSAS instance.

The AlwaysOn tab displays information about your AlwaysOn environment, including Overview Diagrams with complete topology, Historical Charting, and a detailed level Gridiviews. When you monitor the SQL Server instance hosting the Primary Replica of an availability group with SQL Sentry, the entire topology of that availability group will be displayed as part of the Overview Area. Unwatched instances hosting replicas are displayed with a gray background.

This tab is not displayed for an Azure SQL Database target or SSAS instance.

The Top SQL tab contains information about your high impact worst performing stored procedures, batches, and statements. Use the Top SQL tab to quickly identify and correct the worst performing offenders. The information Top SQL captures can be based on any combination of duration, cpu, reads, or writes. From the Top SQL tab you can use the View button in the Plan column to open a Plan Explorer session for any captured query.

The Query Plans tab lists all plans collected for Top SQL events for the specified date range. This provides a detailed chronology of all query plan
changes, so you can go back to any point in time to see when a plan change may have led to a query performance problem.

This tab is not displayed for an Azure SQL Database target or SSAS instance.

The Blocking SQL tab contains a hierarchical view of blocking process chains. You can use the Blocking SQL tab to identify blocks happening in real-time and kill the offending process, or you can view historical information about any captured blocks. The information Blocking SQL captures is based on a minimum block duration and can be adjusted as needed.

This tab is not displayed for an Azure SQL Database target or SSAS instance.

The Deadlocks tab contains information about deadlocks happening in your environment. Use the Deadlocks tab to identify and correct Deadlocks in your environment. Deadlock captures can be turned off for individual Instances as needed.

This tab is not displayed for an Azure SQL Database target or SSAS instance.

The Quick Traces tab can be used to execute a new Quick Trace or view historical data about past Quick Traces. Use Quick Traces to gather full statement and process level details while the problem is still occurring without lengthy trace setup and execution. You can also automate Quick Traces to take place when specific conditions are detected via the Run Quick Trace Action.

This tab is not displayed for an Azure SQL Database target or SSAS instance.

WATCHING A SQL SERVER WITH SQL SENTRY

The following features are enabled with SQL Sentry:

PERFORMANCE ANALYSIS TABS

- **Jump To Calendar** context items are made available from various points allowing you to navigate directly to a calendar view for the active date range showing all associated Performance Analysis events (Top SQL, Blocking SQL and Deadlocks) alongside Event Calendar events such as SQL Agent Jobs, and Windows Tasks.

NAVISOR

- New event source nodes are added under SQL Servers for Top SQL, Blocking SQL and Deadlocks, enabling calendars and other functions for these sources.

EVENT VIEW

- The Event Sources tab has new options for Top SQL, Blocking SQL and Deadlocks to control whether or not these sources are displayed in the view.

CALENDAR

- View Top SQL, Blocking SQL, and Deadlocks on SQL Server, Device, and Custom View calendars.
- Jump To Performance Analysis context items on all calendar events will take you to the
selected tab of Performance Analysis preset to the current calendar date range.

- Opening a Top SQL, Block, or Deadlock event on the calendar will take you directly into Performance Analysis with that event selected and expanded.

PERFORMANCE ANALYSIS CONDITIONS, ACTIONS & SETTINGS

- Performance Analysis conditions are available in the Conditions Pane for SQL Server Instances:
  - Blocking SQL
  - Blocking SQL: Output Content Match
  - Blocking SQL: Runtime Threshold Max
  - Deadlock
  - Deadlock: Output Content Match
  - Top SQL
  - Top SQL: Output Content Match
  - Top SQL: Runtime Threshold Max

- The Run Quick Trace action is available for certain SentryOne conditions. This action will execute and save a Quick Trace when the condition is triggered, including:
  - Event Chain Node: Runtime Threshold Max
  - Event Chain: Runtime Threshold Max
  - Performance Counter: Threshold Max
  - Performance Counter: Threshold Min
  - SQL Server Agent Job: Block
  - SQL Server Agent Job: Retry
  - SQL Server Agent Job: Runtime Threshold Max
  - SQL Server: Blocking SQL
  - SQL Server: Blocking SQL: Runtime Threshold Max

- Performance Analysis settings are available from the Settings pane:
  - Blocking SQL
  - Blocking SQL Source
  - Deadlocks Source
  - Top SQL
  - Top SQL Source

WIN SENTRY

Win Sentry supports monitoring Windows Targets, even if the Target does not have an active SQL Server installation. This gives you the ability to independently monitor any Windows Target. For instance, you may have a need to independently monitor a web server. In this case, Win Sentry delivers a complete historical record of which processes are consuming which resources.

Win Sentry includes the Performance Analysis Dashboard and the Processes tab. If you are
familiar with SQL Sentry you will recognize the Windows charts from the left half of the Dashboard. Several enhancements have been made to the Dashboard with Windows only monitoring in mind. The System Memory and CPU Usage charts now contain visual representations for several different well-known process groups, including groups for SSRS, SSIS, and IIS. If you have a specialized group of applications, Win Sentry even gives you the ability to define your own well-known process groups.

The Processes tab is also included as part of Win Sentry. This tab contains a gridview of all the processes which you are collecting information about, including related metrics. By default, processes are shown in their well-known process groups, giving you a complete picture of how application groups are consuming resources within your environment. Processes are also auto-correlated with related services.

**ADDING A WINDOWS TARGET**

In order to monitor a server with Win Sentry you will first need to add the Windows target to your installation. You may add a Windows Target using the Add Target command found in the right-click context menu of the following Navigator pane nodes: All Targets (root node), Site node, or Group node. You may also add a target through the File menu.

From the Add Target dialog you will choose Windows Computer from the Target Type dropdown menu.

> Note: You may monitor the individual Windows machines which are part of a Windows Cluster, but Win Sentry is not cluster aware.

9.1 Dashboard

**Applies to:** SQL Sentry, Win Sentry, BI Sentry, V Sentry, and DB Sentry
*Some features may not be available to all target types. Please read further for more detail.

The Performance Analysis Dashboard provides an overview of different targets and instances, such as Windows and SQL Server. The dashboard displays the most important metrics used for determining health and load. Performance data is organized into two vertical panes, split into smaller sections dependent upon the type of dashboard you are viewing. The Dashboard for a Windows based target with a SQL Server instance will show Windows operating system specific metrics in the left pane, and SQL Server or SSAS metrics in the right pane.

Please see the Performance Metrics section of the User Guide for a detailed description of the various performance metrics displayed on the Dashboard.

Metrics for all SQL Server or SSAS instances on the server are integrated into the Windows Network: Total, CPU Usage: Total, and System Memory charts, so you can quickly determine exactly how much of each of these resources is being used by each instance, for any point in time. When hovering over colored sections of these charts the corresponding instance will be displayed. Note that each instance has a unique color assigned to it that remains constant across all charts.

For performance reasons, the dashboard never queries the remote server directly -- all data is
retrieved from the SentryOne database after it has been collected by the SentryOne Monitoring Service. This means that there may be a slight delay between the time the performance data was collected and the time it's displayed on the dashboard, although this is typically no more than a few seconds.

When in real time mode (auto-refresh is enabled), the charts and gauges will update automatically approximately every 10 seconds to show the values for the last collected sample.

**DASHBOARD DISPLAY MODES**

The Dashboard has two display modes: **Sample** and **History**. The display mode can be changed at any time by toggling the mode toolbar button. Note the graphic and the text will change when toggled, and will always indicate the mode that will become active when you click the button, versus the currently active mode.

**Sample** mode uses various bar charts and digital gauges to show activity for a specific time, or sample time. This mode includes certain metrics that aren't available in History mode, such as *Network Activity by Adapter, CPU Usage by individual Processor, and SQL Server waits by specific wait class.*

**History** mode uses line and stacked area charts to show you how the Windows server and SQL Server have been performing over any time range. The range can be changed by adjusting the **Start** and/or **End Times** on the toolbar, then clicking the **Go button**. There is no restriction on the size of the range, although the resolution of the data displayed will change, growing less detailed as you zoom out. For example, for any range size of 30 minutes or less you will be viewing the actual raw performance data, and if you zoom out to an 8 hour range, 10-minute averages will be displayed. For more information see the [Dashboard Resolutions/Retention topic.](#)

**CUSTOMIZING SQL SERVER WAITS CHART**

You can customize the SQL Server Waits historical chart by adding an additional category to the PerformanceAnalysisWaitTypeCategory table, then assigning counters in the PerformanceAnalysisWaitType table to the newly created category.

For example, you could create a new category for parallelism by using the following query.

**New Category Creation**

```sql
INSERT INTO [SentryOne].[dbo].[PerformanceAnalysisWaitTypeCategory] (Name, SortOrder, Enabled, RGBColor) VALUES ('Parallelism', 6, 1, '#bb7fd1')
```

After creating the Parallelism category, you can add a counter to it.

**Adding a Counter to the Category**

```sql
UPDATE [SentryOne].[dbo].[PerformanceAnalysisWaitType] SET WaitTypeCategoryID = 7 WHERE Name = 'CXPACKET'
```

**INTERACTING WITH THE DASHBOARD**
Changing the active SQL Server/SSAS instance

You can change the SQL Server or SSAS instance using the **Instance** dropdown control in the top middle of the screen. The metrics in the right pane will reload, but the active date range and other options will be maintained.

The following functions are available when in **History mode**:

**Toolips**

Hover over any point on the chart to see tooltips with specifics about the sample, including the sample instance name, time, and value.

**Sample Selection**

Clicking on a history chart will present a vertical red line at that position called a cursor. The selected time is indicated by the **Sample Pos** box at the top right. The cursor is synchronized across all charts so that you can easily correlate what was happening with one performance metric with what was happening with other related metrics at that same time.

After selecting a sample with the cursor, if you toggle the Mode button, the detail data for the selected time will be loaded in sample mode, allowing you to see specifics for that sample that aren't available in history mode. For example, if you click on a spike on the SQL Server Waits chart and change to sample mode, you can see the specific wait classes were responsible for the spike, whereas in history mode only general wait categories are displayed.

**Click and Hold**

If you **click and hold** a sample, after a short time the cursor will change color, and when you release the mouse you'll be presented with the a subset of the **Jump To** context items used for range selection (below), allowing you to jump directly to a different tab and view only operations that overlapped the currently selected sample time.

**Range Selection**

Click on a history chart and drag the cursor to select a range. The **Range Start** and **Range End** boxes will update, and when you release the mouse button context items will be presented to allow you do choose whether to **Zoom Into** the selected range, or **Jump To** another part of the application using the selected range. For example, if you select a range that includes a spike in network activity, then select **Jump To -> Top SQL**, the **Top SQL** tab will be displayed and loaded only with queries that ran during that range, enabling you to determine which queries may have been responsible for the network spike.

---

**Context Menus**

**Note**: Not all target dashboards will include all the context menus described below. For example, since Quick Trace is not available on Azure SQL Database the context menu for that
feature will not be shown.

Sample Mode Context Menus
- Show History Mode - changes the view to history mode.
- Quick Trace - run a Quick Trace.
- Show Windows Processes - displays Windows processes running on the server.

History Mode Context Menus
- Show Sample Mode - changes the view to sample mode.
- Quick Trace - run a Quick Trace.
- Show Windows Processes - displays Windows processes running on the server.
- Zoom In - Zoom into the selected range.
- Jump To - Jump to another Performance Analysis tab or Event Calendar with the selected date range.

Controls

Sample Mode
Network
Total - displays network activity as an aggregate across all adapters, with SQL Server/SSAS activity integrated.
Adapters - displays activity by the individual network adapter, enabling you to spot adapter-specific saturation issues.

History Mode
Disk IO
Total - displays average latency across all physical disks.
By Disk - displays actual latency for a specific physical disk.

Database IO
Total - displays average latency across all database files.
By Database file - displays actual latency for a specific database file.

USING QUICK TRACE™
Quick Trace gives you an instant snapshot of what all of the processes on a SQL Server are currently doing. Trace and process-level data is collected for a specified interval and then automatically correlated, effectively combining the functionality of Profiler and Activity Monitor together with a single click. You launch Quick Trace by clicking the Quick Trace toolbar button, by right-clicking a Target in the Navigator pane, or directly from the dashboard by right-clicking any chart. For example, if you observe a spike in activity when in real time mode, you can immediately launch Quick Trace and see exactly which SQL Server processes or applications are responsible for the spike.

Quick Trace is not supported on an Azure SQL Database target.
See the Quick Traces topics for more details.

VIEWING WINDOWS PROCESS ACTIVITY
You can see metrics for all Windows processes using **Show Windows Processes** function directly from SentryOne, to avoid having to open a Remote Desktop connection to a server and open Windows Task Manager. The **Process Activity** window, which is similar to Task Manager, will load listing all active operating system processes, sorted descending by CPU by default. You can launch the Process Activity window using the Show Windows Processes toolbar button, by right-clicking a Target in the Navigator pane, or directly from the dashboard by right-clicking any chart.

Like other SentryOne grids, to eliminate background noise, rows with zero values in the sort column are hidden by default. You can unhide them by unchecking the associated option at the bottom of the screen.

Viewing Windows Process Activity is not supported on an Azure SQL Database target.

**Note:** Administrative access to the target server is required in order to launch the Process Activity viewer.

**EVENT OVERLAYS**

Specific events taking place in your environment can also be overlaid on the Performance Analysis Dashboard. This feature can be accessed via the **Jump To Dashboard** option from a number of places within the SentryOne Client which display a range of historical events, including any: **Event Calendar**, **Top SQL tab**, **Blocking tab**, or **QuickTraces tab**.

When you use the **Jump To Dashboard** feature, events are overlaid on the x-axis for each chart. The start of each distinct event will be a represented with a green circle, with the event’s duration represented with a pink line, ending in a pink dot. Hovering over an event will display a tooltip about that event.

- **If the selected time range is not large enough, events will be represented simply by a green circle. If this is the case, changing the time range to a smaller interval will preserve the overlay of the event, and may offer a better view of the impact that each event had.**

**From the Event Calendar**

When using the **Jump To Dashboard** feature from the **Event Calendar** any events which are highlighted via the **Calendar Highlighting** feature will be overlaid on the Dashboard.

**From the Top SQL tab**

When using the **Jump To Dashboard** feature from the **Top SQL tab**, if the Show Totals mode is enabled all events belonging to the selected group will be overlaid on the Dashboard. If the Default mode is enabled only the selected event will be overlaid on the Dashboard.

**From the Blocking or QuickTraces tab**

When using the **Jump To Dashboard** feature from the **Blocking or QuickTraces tab** the selected event will be overlaid on the Dashboard.

9.1.1 Dashboard Resolutions/Retention

**Applies to:** SQL Sentry, Win Sentry, BI Sentry, V Sentry, and DB Sentry

**DASHBOARD RESOLUTIONS**

The **Performance Analysis Dashboard** will display metrics at different resolution levels,
depending on the historical time range which is selected in the toolbar. Whenever you are viewing
the Dashboard in either Sample mode, or in History mode ranges of 30 minutes or less, the raw
samples collected (detail data) will be displayed. This detail level data has a collection frequency of
anywhere from 10 seconds to 1 minute, depending on the metric. As you expand the time range
you are viewing, averages over the indicated resolution interval are shown.

<table>
<thead>
<tr>
<th>Time Range</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 &lt;= 30 min</td>
<td>Detail Data</td>
</tr>
<tr>
<td>30 min &lt;= 8 hrs</td>
<td>2 min</td>
</tr>
<tr>
<td>3 hrs 20 min &lt;= 36 hrs</td>
<td>10 min</td>
</tr>
<tr>
<td>12 hrs &lt;= 5 days</td>
<td>30 min</td>
</tr>
<tr>
<td>2 days &lt;= 20 days</td>
<td>2 hrs</td>
</tr>
<tr>
<td>3 days 8 hrs &lt;= 40 days</td>
<td>4 hrs</td>
</tr>
<tr>
<td>10 days &lt;= 1 year</td>
<td>1 day</td>
</tr>
<tr>
<td>20 days &lt;= 2 years</td>
<td>2 day</td>
</tr>
<tr>
<td>30 days &lt;= 3 years+</td>
<td>3 day</td>
</tr>
</tbody>
</table>

**HISTORICAL DATA RETENTION**

Performance Analysis uses the SentryOne Database to store all of the performance data it collects.
Performance data is rolled up or averaged based on the age of the data. Once data becomes old
enough to move into the next range it will be represented by an average over the time. The
default settings for data retention for the different resolutions are listed below:

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Retention Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail Data</td>
<td>3 days</td>
</tr>
<tr>
<td>2 min</td>
<td>5 days</td>
</tr>
<tr>
<td>10 min</td>
<td>10 days</td>
</tr>
<tr>
<td>30 min</td>
<td>20 days</td>
</tr>
<tr>
<td>2 hour</td>
<td>40 days</td>
</tr>
<tr>
<td>4 hour</td>
<td>60 days</td>
</tr>
<tr>
<td>1 day</td>
<td>1 year</td>
</tr>
<tr>
<td>2 day</td>
<td>2 years</td>
</tr>
<tr>
<td>3 day</td>
<td>3 years</td>
</tr>
</tbody>
</table>

**EXAMPLE 1:** A DBA comes into work on Monday and wants to investigate a performance related
issue that happened late Friday afternoon. Since detail data is retained for 3 days (72 hours) they
will be able to access it using the most granular detail resolution (0 <= 30 min.).

**EXAMPLE 2:** A DBA wants to compare performance data for the last two months (60 days). The
data will need to be compared using the values for the 4 hour resolution, since that is the
resolution available at 60 days by default.

**NOTE:** The data retention period can be customized for specific needs. See the SentryOne Data
Capacity Planning article for more information.
9.1.2 Baselining

Applies to: SQL Sentry, Win Sentry, BI Sentry, V Sentry, and DB Sentry

Baselines can be created and managed from the History view of the Performance Analysis Dashboard. There are two distinct baseline types available. The first type of baseline is the Predefined type. SentryOne includes several predefined baselines which are based on the historical metrics gathered in your monitored environment. The second type of baseline is the Custom type. SentryOne also allows you to define your own custom baselines.

Once a baseline has been selected, it is overlaid on the various Performance Analysis Dashboard charts, giving you “at a glance” information that can help you to manage change in your monitored environment. Custom baselines can also be used in Advisory Conditions, allowing the comparison of baseline values to live metrics, as they are captured.

At this time you cannot display a baseline created by one type of target overlaid on a different type of target. For example, creating a baseline against a SQL Server Instance running on a Windows Target cannot be then overlaid on an Azure SQL Database dashboard.

PREDEFINED BASELINES

Predefined baselines are available without any custom configuration. They can be accessed from the Baseline drop-down menu on the Dashboard while in History view. The different predefined baselines are available based on the historical time range you are viewing on the Dashboard. Reference the table below for more information:

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Time Range Availability</th>
<th>Range Option Availability (detail data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Period</td>
<td>Is always available. Corresponds to the immediate preceding timeframe.</td>
<td>Yes</td>
</tr>
<tr>
<td>Previous Day</td>
<td>0 &lt;= 24 hrs</td>
<td>Yes</td>
</tr>
<tr>
<td>Previous Week</td>
<td>0 &lt;= 168 hrs</td>
<td>Yes</td>
</tr>
<tr>
<td>Previous Month</td>
<td>36 &lt;= 720 hrs</td>
<td>No</td>
</tr>
<tr>
<td>Previous Quarter</td>
<td>480 &lt;= 2160 hrs</td>
<td>No</td>
</tr>
<tr>
<td>Previous Year</td>
<td>960 &lt;= 8760 hrs</td>
<td>No</td>
</tr>
</tbody>
</table>

AVERAGE AND RANGE MODES

It is important to understand that predefined baselines are based on the historical data collected in your monitored environment. The granularity of the data-points which make up a baseline is entirely dependent on the time range you are viewing.

AVERAGE

By default, predefined baselines are shown with data-points which show the average value over that sample period.

RANGE

When viewing certain predefined baselines, within a time range which has a resolution of detail data, an additional option will be available, allowing you to view that baseline with a Range of values (reference the Range Option Availability column in the chart above). This range includes both a minimum and maximum value observed during the respective sample period.
CUSTOM BASELINES

Custom baselines can be created on demand via the Dashboard user interface while in History view. To create a custom baseline, first left-click on any Dashboard chart and drag to highlight your desired timeframe. Next, select the Create Baseline option available from the context menu. The Create Baseline form will be displayed.

CREATE BASELINE FORM

The Create Baseline form contains information about all of the metrics which are captured as part of a custom baseline. See below for column explanations.

<table>
<thead>
<tr>
<th>Metric</th>
<th>The name of the metric captured in the baseline.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Defines which metric in the Chart will be used when displaying the baseline. Note: This can be changed at any time.</td>
</tr>
<tr>
<td>Average</td>
<td>Average is the value which will be used for that particular metric in the baseline. This controls both the actual line displayed on the respective chart and the value used for comparison purposes in Advisory Conditions. Average is calculated based on the observed values for the particular metric within the baseline’s time range. To define custom values to be used with the baseline, simply edit the pre-calculated Average value.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum value captured during the baseline’s time range.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum value captured during the baseline’s time range.</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>The standard deviation for the values captured during the baseline's time range.</td>
</tr>
</tbody>
</table>

USING CUSTOM BASELINES IN ADVISORY CONDITIONS

The values defined for metrics which make up a custom baseline can be used in Advisory Conditions. To access a baseline in a Advisory Condition, you will need to define the Advisory Condition at that same Instance level at which the baseline is defined. For example, if you create a baseline on the server named LONDON.UK.COM, in order to access the baseline in an Advisory Condition, you will need to create the Advisory Condition from the context of the LONDON.UK.COM server in the Navigator pane. For an in-depth look at Advisory Conditions see the Advisory Conditions topic.

BASELINES AND PERFORMANCE COUNTER MAPPING

When using baselines in Advisory Conditions you will need to reference the following tables which includes the mapping between the metrics captured in baselines and their associated friendly Performance Counter names.

SQL Server Baseline Mapping

<table>
<thead>
<tr>
<th>Chart</th>
<th>Metric</th>
<th>Associated Exposed Performance Counter (Advisory Condition format)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup/Restore</td>
<td>Backup MB/sec</td>
<td>SQL Server Database Engine: SQL Server: Databases: Backup/Restore MB/sec</td>
</tr>
<tr>
<td>Cache Hit Ratios</td>
<td>Page Life Expectancy (sec)</td>
<td>SQL Server Database Engine- Buffer Node - PLE : Equals : &quot;Node Name&quot;</td>
</tr>
<tr>
<td></td>
<td>SQL Plans</td>
<td>SQL Server Database Engine Plan Cache Hit Ratio: Equals: SQL Plans</td>
</tr>
<tr>
<td></td>
<td>Object Plans</td>
<td>SQL Server Database Engine Plan Cache Hit Ratio: Equals: Object Plans</td>
</tr>
<tr>
<td>Chart</td>
<td>Metric</td>
<td>Associated Exposed Performance Counter (Advisory Condition format)</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Checkpoint pgs</td>
<td>Lazy writes</td>
<td>Checkpoint pages/sec</td>
</tr>
<tr>
<td></td>
<td>Lazy writes/sec</td>
<td>Lazy writes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>CPU Usage</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total CPU Usage %</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Database Latency</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total ms/Read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total ms/Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Disk IO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total ms/Read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total ms/Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td>**Faults (Read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Faults: Read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write Faults: Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td>**Key lookups</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Key lookups/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Forwarded records/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Log Flushes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Log Flushes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Network In</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>In %</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Network out</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Out %</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Pages</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page reads/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page writes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SQL Activity</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Batches/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transactions/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compiles/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Recompiles/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SQL Server Memory</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total SQL Memory Usage (MB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SQL Server Waits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total Wait Time (ms)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>System Memory</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total Memory Usage (MB)</td>
</tr>
</tbody>
</table>

**SSAS Baseline Mapping**

<table>
<thead>
<tr>
<th>Chart</th>
<th>Metric</th>
<th>Associated Exposed Performance Counter (Advisory Condition format)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Usage</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total CPU Usage %</td>
<td>Windows: Processor Information: % Processor Time: Total</td>
</tr>
<tr>
<td>Chart</td>
<td>Metric</td>
<td>Associated Exposed Performance Counter (Advisory Condition format)</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Disk IO</td>
<td>Total ms/Read</td>
<td>Windows: PhysicalDisk: ms/Read</td>
</tr>
<tr>
<td></td>
<td>Total ms/Write</td>
<td>Windows: PhysicalDisk: ms/Write</td>
</tr>
<tr>
<td>Faults (Read</td>
<td>Write)</td>
<td>Faults: Read</td>
</tr>
<tr>
<td></td>
<td>Faults: Write</td>
<td>Windows: Memory: Faults: Write</td>
</tr>
<tr>
<td>Network In</td>
<td>In %</td>
<td>Windows: Network Interface: Recieved %: Total</td>
</tr>
<tr>
<td>Network out</td>
<td>Out %</td>
<td>Windows: Network Interface: Sent %: Total</td>
</tr>
<tr>
<td>SSAS Activity: Rows Transferred</td>
<td>Query rows sent/sec</td>
<td>SSAS: Storage Engine Query rows sent/sec</td>
</tr>
<tr>
<td></td>
<td>Processing rows read/sec</td>
<td>SSAS: Processing: Processing rows read/sec</td>
</tr>
<tr>
<td>SSAS Activity: Threads Formula Engine</td>
<td>Query pool busy threads</td>
<td>SSAS: Threads: Query pool busy threads</td>
</tr>
<tr>
<td></td>
<td>Query pool job rate</td>
<td>SSAS: Threads: Query pool job rate</td>
</tr>
<tr>
<td></td>
<td>Query pool job queue length</td>
<td>SSAS: Threads: Query pool job queue length</td>
</tr>
<tr>
<td>SSAS Activity: Threads Storage Engine</td>
<td>Busy</td>
<td>SSAS: Threads: Processing pool busy I/O job threads</td>
</tr>
<tr>
<td></td>
<td>Processing pool job rate</td>
<td>SSAS: Threads: Processing pool I/O job completion rate</td>
</tr>
<tr>
<td></td>
<td>Queued</td>
<td>SSAS: Threads: Processing pool I/O job queue length</td>
</tr>
<tr>
<td>SSAS General : Avg Time</td>
<td>Total Wait Time (ms)</td>
<td>SSAS Trace Waits: Total Wait Time (ms)</td>
</tr>
<tr>
<td>SSAS General: MDX</td>
<td>Cells calculated/sec</td>
<td>SSAS: MDX : Cells calculated/sec</td>
</tr>
<tr>
<td></td>
<td>Calculation covers/sec</td>
<td>SSAS: MDX : Calculation covers/sec</td>
</tr>
<tr>
<td></td>
<td>Sonar subcubes/sec</td>
<td>SSAS: MDX : Sonar subcubes/sec</td>
</tr>
<tr>
<td></td>
<td>Recomputes/sec</td>
<td>SSAS: MDX : Recomputes/sec</td>
</tr>
<tr>
<td></td>
<td>NON EMPTY unoptimized/sec</td>
<td>SSAS: MDX : NON EMPTY unoptimized/sec</td>
</tr>
<tr>
<td></td>
<td>NON EMPTY for calculated members/sec</td>
<td>SSAS: MDX : NON EMPTY for calculated members/sec</td>
</tr>
<tr>
<td>SSAS General: Processing</td>
<td>Processing rows written/sec</td>
<td>SSAS: Processing: Processing rows written/sec</td>
</tr>
<tr>
<td></td>
<td>Index rows created/sec</td>
<td>SSAS: Proc Indexes: Index rows created/sec</td>
</tr>
<tr>
<td></td>
<td>Aggregation rows created/sec</td>
<td>SSAS: Proc Aggregations: Aggregation rows created/sec</td>
</tr>
<tr>
<td>SSAS Mem Usage (MB)</td>
<td>Total Cleaner Memory (MB)</td>
<td>SSAS: Memory: Total Cleaner Memory (MB)</td>
</tr>
<tr>
<td></td>
<td>Total Category Memory (MB)</td>
<td>SSAS: Memory: usage by Category: Total memory Usage (MB)</td>
</tr>
<tr>
<td>SSAS Memory: Cache Activity</td>
<td>Cache inserts/sec</td>
<td>SSAS: Cache: Cache inserts/sec</td>
</tr>
<tr>
<td></td>
<td>Cache evictions/sec</td>
<td>SSAS: Cache: Cache evictions/sec</td>
</tr>
<tr>
<td></td>
<td>Flat cache inserts/sec</td>
<td>SSAS: MDX: Flat cache inserts/sec</td>
</tr>
<tr>
<td></td>
<td>KB added/sec</td>
<td>SSAS: Cache: KB added/sec</td>
</tr>
<tr>
<td></td>
<td>KB shrunk/sec</td>
<td>SSAS: Memory: KB shrunk/sec</td>
</tr>
<tr>
<td>Chart</td>
<td>Metric</td>
<td>Associated Exposed Performance Counter (Advisory Condition format)</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>SSAS Memory: Cache Lookups</td>
<td>Dimension cache hits/sec</td>
<td>SSAS: Cache: Lookups/sec</td>
</tr>
<tr>
<td></td>
<td>Measure group cache lookups/sec</td>
<td>SSAS: Storage Engine Query: Measure group cache lookups/sec</td>
</tr>
<tr>
<td></td>
<td>Calculations cache lookups/sec</td>
<td>SSAS: Storage Engine Query: Calculations cache lookups/sec</td>
</tr>
<tr>
<td></td>
<td>Flat cache lookups/sec</td>
<td>SSAS: Storage Engine Query: Flat lookups/sec</td>
</tr>
<tr>
<td>SSAS Storage: Current Partitions</td>
<td>Indexes</td>
<td>SSAS: Proc Indexes: Current partitions (indexes)</td>
</tr>
<tr>
<td></td>
<td>Aggregations</td>
<td>SSAS: Proc Aggregations: Current partitions (aggregations)</td>
</tr>
<tr>
<td>SSAS Storage: File Queries</td>
<td>Total queries from file/sec</td>
<td>SSAS: Storage Engine Query: Total queries from file/sec</td>
</tr>
<tr>
<td></td>
<td>Aggregation hits/sec</td>
<td>SSAS: SSAS Trace: Aggregation hits/sec: Total</td>
</tr>
<tr>
<td>SSAS Storage: File Read KB</td>
<td>Total System File Read KB</td>
<td>Windows:System:Total System read KB/Sec</td>
</tr>
<tr>
<td>SSAS Storage: File Write KB</td>
<td>Total System File Write KB</td>
<td>Windows:System:Total System write KB/Sec</td>
</tr>
<tr>
<td>System Memory</td>
<td>Total Memory Usage (MB)</td>
<td>Windows: Memory: Total used Memory (MB)</td>
</tr>
</tbody>
</table>

**PERFORMANCE METRICS**

This topic covers the various performance metrics displayed by the Performance Analysis Dashboard and Performance Analysis Overview, and how to interpret different metric values and combinations of values across different metrics.

**Windows Metrics**
- Network
  - Total network utilization
  - Utilization by SQL Server/SSAS instance
  - Utilization by network adapter
  - Network output queue length
- CPU Usage
  - Total processor time
  - User time by CPU
  - Kernel time by CPU
  - % Guest Runtime
  - vCPU Wait Time
- Processor time by SQL Server/SSAS process
- Context Switches
- Processor queue length
- System Memory
- Memory by SQL Server/SSAS instance
- Other processes memory
- File Cache memory
- Available Memory
- Ballooned memory
- Read Faults
- Write Faults
- Page file usage
- Disk I/O
- Read latency by physical disk
- Write latency by physical disk
SQL Server Metrics
SQL Server Activity
User Connections
Blocked Processes
Batches
Transactions
Compiles
Recompiles
Key lookups
Forwarded records
Backup MB/sec
Send Queue / Redo Queue
SQL Server Waits
CPU % of Total Waits
Wait Time: by Category
Wait Time: by Class
SQL Server Memory
Buffer cache size
Plan cache size (SQL)
Plan cache size (Objects)
Other cache size
Plan cache hit ratio (SQL)
Plan cache hit ratio (Objects)
Page reads
Page writes
Page life expectancy
Memory grants pending
Database IO
Read latency by database file
Write latency by database file
Checkpoint pages
Lazy Writes
Log Flushes
Hyper-V Host Metrics
CPU Usage (VM)
Utilization by Virtual Machine Guest
Usage time by Guest
vCPU Root usage time
% Guest Runtime
vCPU Wait Time
System Memory (VM)
Memory by Virtual Machine Guest
Disk IO (VM)
IOPS (Read)
IOPS (Write)
Read Throughput
Write Throughput
Azure SQL Database Metrics
DTU Usage
Total DTU %
CPU
Data IO
Log IO
Memory Usage
Allocated Memory Usage
Database Size
Size in GB
SQL Server Activity
User Connections
Blocked Processes
Transactions
## WINDOWS METRICS

<table>
<thead>
<tr>
<th>Section</th>
<th>Performance Metric</th>
<th>Type</th>
<th>Description</th>
<th>Modes</th>
<th>Warning Range</th>
<th>Critical Range</th>
<th>May correlate with</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Total network utilization</td>
<td>%</td>
<td>The total combined % utilization for all adapters on the server.</td>
<td>S, H</td>
<td>&gt;10%</td>
<td>&gt;20%</td>
<td>CPU Usage: Kernel time</td>
</tr>
<tr>
<td>Network</td>
<td>Utilization by SQL Server/SSAS instance</td>
<td>%</td>
<td>The % utilization for each instance is overlaid on top of the total % utilization, enabling you to see how much of the network traffic is related to SQL Server, SSAS, or other processes on the server.</td>
<td>S, H</td>
<td>&gt;3</td>
<td>&gt;5</td>
<td>CPU Usage: Kernel time</td>
</tr>
<tr>
<td>Network</td>
<td>Utilization by network adapter</td>
<td>MB/sec</td>
<td>Select the Adapters radio group item to view activity for up to 4 network adapters on the server. This enables you to spot adapter-specific saturation issues.</td>
<td>S</td>
<td>&gt;3</td>
<td>&gt;5</td>
<td>CPU Usage: Kernel time</td>
</tr>
<tr>
<td>Network</td>
<td>Network output queue length</td>
<td>Last value</td>
<td>The length of the network output packet queue, in packets. This is the total value across all adapters on the machine. A sustained value of &gt;3 may indicate a network bottleneck.</td>
<td>S</td>
<td>&gt;3</td>
<td>&gt;5</td>
<td>Network activity</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>Total processor time</td>
<td>%</td>
<td>The total processor time % across all processors on the server. A sustained value greater than 80% generally indicates a CPU bottleneck.</td>
<td>S</td>
<td>&gt;3</td>
<td>&gt;5</td>
<td>--</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>User time by CPU</td>
<td>%</td>
<td>The user time % for each individual CPU. Represented by the color green. If user time for individual CPUs are at or near 100% for a sustained period, SQL Server or other applications may not be parallelizing effectively.</td>
<td>S, H</td>
<td>&gt;85%</td>
<td>&gt;95%</td>
<td>SQL Server: Cycles, recompiles, batch, transactions, key leakage, CPU wait time, Querier parallelism, truth joins, sort operations, Index rebuilds &amp; defragmenting Consistency checks, SSAS: Formula engine time, Cells calculated, Recalculations, Flat cache lookups, Aggregation row created</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>Kernel time by CPU</td>
<td>%</td>
<td>The kernel (or privileged) time % for each individual CPU. Represented by the color red. Kernel time should typically be &lt;4% total, and &lt;2% of user time. If kernel time for individual CPUs are high for a sustained period, a driver may be responsible. For example, a network driver may cause high kernel time, and it may be isolated to specific CPUs, in which case it may correlate with network activity.</td>
<td>S</td>
<td>&gt;3</td>
<td>&gt;5</td>
<td>Network activity</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>Processor time by SQL Server/SSAS instance</td>
<td>%</td>
<td>CPU time related to SQL Server and SSAS processes is overlaid on top of the total CPU series. Note that there is only one SQL Server chart series and it represents all SSAS instances on the server combined. Use SQL Server Quick Trace to determine which queries are utilizing the most CPU resources.</td>
<td>S, H</td>
<td>&gt;3</td>
<td>&gt;5</td>
<td>SQL Server: Cycles, recompiles, batch, transactions, key leakage, CPU wait time, Querier parallelism, truth joins, sort operations, Index rebuilds &amp; defragmenting Consistency checks, SSAS: Formula engine time, Cells calculated, Recalculations, Flat cache lookups, Aggregation row created</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>% Guest Runtime*</td>
<td>%</td>
<td>For the Hyper-V Hypervisor Logical Processor, the average percentage of time guest code is running on an Logical Processor.</td>
<td>S, H</td>
<td>&gt;2.5</td>
<td>&gt;3</td>
<td>--</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>vCPU Wait Time*</td>
<td>%</td>
<td>The % time the guest virtual machine spent waiting for a Host kernel resource.</td>
<td>S, H</td>
<td>&gt;5000 times # of processors</td>
<td>&gt;7500 times # of processors</td>
<td>CPU Usage: Kernel time</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>Context switches</td>
<td>Avg/sec</td>
<td>The number of times that all processors on the computer are switched from one thread to another. Consistently high values can mean that the server is spending too much time switching threads instead of actively running threads.</td>
<td>S</td>
<td>&lt;2000 times # of processors</td>
<td>&gt;3000 times # of processors</td>
<td>--</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>Processor queue length</td>
<td>Last value</td>
<td>The number of threads in the processor queue. If this value goes over 2 per processor, and CPU usage for SQL Server is high, it can be indicative of high complex/recomplex, or a high rate of key lookups, which can often be addressed by covering indexes. Use SQL Server Quick Trace to troubleshoot</td>
<td>S</td>
<td>&gt;3 times the # of processors</td>
<td>&gt;5 times the # of processors</td>
<td>--</td>
</tr>
<tr>
<td>System Memory</td>
<td>Memory by SQL Server/SSAS instance</td>
<td>MB</td>
<td>The amount of physical memory used by each SQL Server and SSAS instance. Important for determining whether available memory is being used effectively, and whether or not there is memory contention between multiple instances on the same server.</td>
<td>S, H</td>
<td>&gt;85%</td>
<td>&gt;95%</td>
<td>--</td>
</tr>
<tr>
<td>System Memory</td>
<td>Memory by SQL Server/SSAS instance</td>
<td>MB</td>
<td>The amount of physical memory used by each SQL Server and SSAS instance. Important for determining whether available memory is being used effectively, and whether or not there is memory contention between multiple instances on the same server.</td>
<td>S, H</td>
<td>&gt;85%</td>
<td>&gt;95%</td>
<td>--</td>
</tr>
<tr>
<td>System Memory</td>
<td>Other processes memory</td>
<td>MB</td>
<td>The amount of physical memory used by all processes on the server other than SQL Server or SSAS. Note that if a SQL Server or SSAS instance isn’t being watched by SentryOne, the memory it uses will be included here.</td>
<td>S, H</td>
<td>&gt;85%</td>
<td>&gt;95%</td>
<td>--</td>
</tr>
<tr>
<td>System Memory</td>
<td>File Cache memory</td>
<td>MB</td>
<td>The amount of physical memory currently allocated to the system file cache. SSAS – SSAS database files may be loaded into and served from the file cache, even if the associated file data doesn’t exist in the SSAS internal caches. For this reason, monitoring the file cache is important to ensure that physical memory is being used effectively, and that memory contention doesn’t occur between the SSAS process, the file cache, and other processes on the server.</td>
<td>S, H</td>
<td>&gt;85%</td>
<td>&gt;95%</td>
<td>--</td>
</tr>
</tbody>
</table>

Backup MB/sec | SQL Server Waits | CPU % of Total Waits | Wait Time: by Category | Wait Time: by Class | SQL Server Memory | Buffer Cache Size | Page Life Expectancy | Database IO | Read Latency by Database File | Write Latency by Database File | Log Flushes | VMWare Host Metrics | CPU | Ready | Memory | Active | Consumed | Overhead | Granted

Memory ballooning allows a physical host to recapture unused memory on a virtual machine, and use it elsewhere. The portion of memory being

---

SentryOne User Guide | 189

---

### SQL Server Metrics

<table>
<thead>
<tr>
<th>Section</th>
<th>Performance Metric</th>
<th>Type</th>
<th>Description</th>
<th>Modes</th>
<th>Warning Range</th>
<th>Critical Range</th>
<th>May correlate with</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Activity</td>
<td>User Connections</td>
<td>Last value</td>
<td>The total number of connections established to the SQL Server. Details for each connection can be viewed by querying the <code>sys.dm_connections</code> DMV.</td>
<td>S, H</td>
<td>&lt;100</td>
<td>&lt;</td>
<td>0-100</td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Blocked Processes</td>
<td>Last value</td>
<td>The total number of blocked processes detected on the SQL Server. The Dashboard is designed to give you &quot;at a glance&quot; information about the number of blocked processes. The Blocking SQL tab should be used to identify blocking or resource usage issues.</td>
<td>S, H</td>
<td>&lt;100</td>
<td>&lt;</td>
<td>0-100</td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Batches</td>
<td>Avg/sec</td>
<td>The total number of bulk operations (insert, update, delete) completed per second.</td>
<td>S, H</td>
<td>&lt;100</td>
<td>&lt;</td>
<td>0-100</td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Transactions</td>
<td>Avg/sec</td>
<td>The total number of transactions per second across all databases on the server. A transaction is either a user-defined batch statement surrounded by a BEGIN TRAN and an END TRAN, or an individual DML statement (insert, update, or delete).</td>
<td>S, H</td>
<td>&lt;100</td>
<td>&lt;</td>
<td>0-100</td>
</tr>
</tbody>
</table>

* Metric will ONLY be visible for targets that are Hyper-V Hosts
** Metric will ONLY be visible for targets that are VM guests whose host is also watched
### SQL Server Activity

#### Compiles

- **Avg/sec**: The number of compiles per second. The value should generally be <15% of initial compiles per second. Also see "Compilations above."  

- **CPU % of Total Waits**: The percentage of all waits which are Signal Waits. Signal wait time is the time a thread has spent waiting on the CPU after being signaled that its resource is available. A high "CPU % of Total Waits" percentage may indicate CPU pressure.  

- **Disk IO**: CPU-intensive operation, especially when large number of rows are involved, since each lookup incurs a random I/O plus additional processing. This often correlates with high CPU usage and page reads. Lookups can be eliminated by using a covering index, adjusting joins to reduce the set so the lookup isn't needed, or using multiple indexes (intersection).  

- **Cache hit ratios**: Generally speaking, total waits of less than 200ms is very good. Between 200ms and 1000ms is average. Greater than 1000ms likely requires some attention to determine where the bottleneck lies. Over 5000ms may indicate severe bottlenecking.

- **SQL Server Memory: Read pages/Sec**: This includes the number of reads performed by the database engine to read pages from disk to memory. This is usually highly correlated with CPU usage, since a high read rate can lead to high CPU usage.

- **SQL Server Memory: CPU Usage**: This shows the percentage of CPU time that is spent on SQL Server. A high percentage can indicate CPU pressure, especially when large number of rows are involved, since each lookup incurs a random I/O plus additional processing. This often correlates with high CPU usage and page reads. Lookups can be eliminated by using a covering index, adjusting joins to reduce the set so the lookup isn't needed, or using multiple indexes (intersection).

- **Disk I/O (Database)**: This is the number of reads and writes performed by the database engine to read and write data to disk. A high number of I/Os can indicate a bottleneck in the disk system, which can lead to increased response times. Also see: [http://blogs.msdn.com/b/craigfr/archive/2006/06/30/652639.aspx](http://blogs.msdn.com/b/craigfr/archive/2006/06/30/652639.aspx)

- **Disk I/O (Cache)**: This is the number of reads and writes performed by the database engine to read and write data to the SQL Server cache. A high number of I/Os can indicate a bottleneck in the cache, which can lead to increased response times. Also see: [http://msdn.microsoft.com/en-us/library/cc966413.aspx](http://msdn.microsoft.com/en-us/library/cc966413.aspx)

- **Disk I/O (System)**: This is the number of reads and writes performed by the operating system to read and write data to disk. A high number of I/Os can indicate a bottleneck in the disk system, which can lead to increased response times. Also see: [http://msdn.microsoft.com/en-us/library/cc966425.aspx](http://msdn.microsoft.com/en-us/library/cc966425.aspx)

- **CPU Usage**: This shows the percentage of CPU time that is spent on SQL Server. A high percentage can indicate CPU pressure, especially when large number of rows are involved, since each lookup incurs a random I/O plus additional processing. This often correlates with high CPU usage and page reads. Lookups can be eliminated by using a covering index, adjusting joins to reduce the set so the lookup isn't needed, or using multiple indexes (intersection).

- **Disk I/O (System)**: This is the number of reads and writes performed by the operating system to read and write data to disk. A high number of I/Os can indicate a bottleneck in the disk system, which can lead to increased response times. Also see: [http://msdn.microsoft.com/en-us/library/cc966425.aspx](http://msdn.microsoft.com/en-us/library/cc966425.aspx)

- **CPU Usage**: This shows the percentage of CPU time that is spent on SQL Server. A high percentage can indicate CPU pressure, especially when large number of rows are involved, since each lookup incurs a random I/O plus additional processing. This often correlates with high CPU usage and page reads. Lookups can be eliminated by using a covering index, adjusting joins to reduce the set so the lookup isn't needed, or using multiple indexes (intersection).
If you hover over a class column, the detailed wait types are shown in a tooltip.

See also: 

SQL Server Memory Buffer cache size MB The current number of the buffer cache, in MB. You want this to be as large as possible for maximum performance, and so as a dedicated SQL Server instance you should consider using most of its SQL Server memory and physical memory.

See also: 

SQL Server Memory Plan cache size (SQL) MB The current number of the cache used for query plans, in MB. This includes ad hoc, ad-hoc parameterized, and prepared plans. A high value in proportion to the buffer cache may indicate query plans aren't being effectively reused.

See also: 

SQL Server Memory Plan cache size (Objects) MB The current number of the cache used for object plans, in MB. This includes bound procedures, functions, and triggers. A high value in proportion to the buffer cache may indicate query plans aren't being effectively reused.

See also: 

SQL Server Memory Other cache size MB The current number of the cache used for all other plans, in MB. This includes bound procedures, extended stored procedures, temporary tables, and table variables. This cache size should be low in proportion to the other plan cache sizes. If it goes over roughly 10% of the object or SQL plan size, further investigation may be needed.

See also: 

SQL Server Memory Plan cache hit ratio (SQL) % The ratio of hits to lookups for the query plan cache. This value should stay above 90%.

See also: 

SQL Server Memory Plan cache hit ratio (Objects) % The ratio of hits to lookups for the object-plan cache. This value should stay above 90%.

See also: 

SQL Server Memory Page reads Avg/sec The average number of buffer page reads from disk per second. Ideally this value should be at or near 0 most of the time. If it is above 0, it means that the data wasn't found in the buffer cache, and so it had to be retrieved from disk. If it's not, page reads correlate with high disk latency, the disk system may not be keeping up.

Note that querying newly created temp tables will also show up as page reads, as well as activity from internal tempdb objects. This includes hash joins, hash aggregates, sort and query spool operations. This means that you can still see high paging from tempdb due to query activity, even though you aren't explicitly using temp tables.

When page reads and page writes correlate closely, it's a strong indicator that it's related to tempdb activity, since pages are being written to disk when the objects are created, then immediately read back in to memory for use by querying operations.

If lazy writes are 0 and track closely with page reads, and page life expectancy <600, it's a strong indicator of memory pressure, since data is being moved out of buffer to make room for new data coming in.

Note that if you're also writing pages, but generally much less than tempdb activity. So if you see high page reads and relatively low page writes and page writes, it's likely memory pressure and not tempdb activity.

See also: 

SQL Server Memory Page writes Avg/sec Page writes can be caused by checkpoints, lazy writes, as well as tempdb activity. To calculate the approximate amount of writes related to tempdb, for any given interval, subtract checkpoints and lazy writes from total page writes.

If high page writes correlate with high latency, the disk system may not be keeping up.

See "Page reads" for more details.

SQL Server Memory Page life expectancy Last value The number of processes waiting for a query workspace memory grants. Ideally this value should be 0 at all times, but it can go above 0 in cases of severe memory pressure.

When it does, RESOURCE_SEMAPHORE waits will also be >0, since this wait is a measure of the time that queries had to wait for memory grants. This type will be visible in the Waits chart tooltips for the Memory class and category.

See also: 

SQL Server Memory Memory grants pending Last value The number of processes waiting for a query workspace memory grants. Ideally this value should be 0 at all times, but it can go above 0 in cases of severe memory pressure.

When it does, RESOURCE_SEMAPHORE waits will also be >0, since this wait is a measure of the time that queries had to wait for memory grants. This type will be visible in the Waits chart tooltips for the Memory class and category.

SQL Server Memory Read latency by database file Avg ms/Read The average time in milliseconds each physical disk read is taking for a particular database file. The top 10 database files (data and transaction log) with the highest latency for the specified date range are shown. In history mode, use the dropdown to determine whether total database latency or a specific database file is shown on the chart.

See "Database IO: Read latency" section for more details and performance ranges.

SQL Server Memory Write latency by database file Avg ms/Write The average time in milliseconds each physical disk write is taking for a particular database file.

See "Database IO: Write latency" section for more details and performance ranges.

SQL Server Memory Checkpoint pages Avg/sec The average number of page writes each second to disk by the checkpoint process. Checkpoints flush all dirty buffer pages for a given database to disk and are a normal part of SQL Server operation. The frequency of checkpoints and volume of checkpoint pages is dictated directly by the "recovery interval" server option. SQL Server uses checkpoints to batch writes to disk, which is generally more efficient. However, if the volume of each checkpoint is too high and you see a correlation with high disk latency, it may indicate that the disk system isn't keeping up.

Checkpoints will sometimes correlate with lower page life expectancy, but only because when pages are flushed from memory, the disk system can drop them more efficiently later on. If you see correlation with high disk latency, it may indicate that the disk system isn't keeping up.

Checkpoints will sometimes correlate with lower page life expectancy, but only because when pages are flushed from memory, the disk system can drop them more efficiently later on. If you see correlation with high disk latency, it may indicate that the disk system isn't keeping up.

This can happen even when the disk is capable of easily sustaining higher throughput, and can lead to abnormal long checkpoints, and blocking of the log writer process.

DBCC TRACER (2002, 2004 – 1) will log many additional details about the checkpoint process to the SQL Server error log, and can be very helpful for troubleshooting checkpoint-related issues.

See also: 

SQL Server Memory Lazy writes Avg/sec The average number of writes per second by the lazy write. The lazy writer periodically scans the buffer and writes pages that have low use counts in order to maintain a certain number of pages on the free list. Ideally, this value should be at or near 0 most of the time. When there is no memory pressure, the lazy writer will generally issue data writes at an even rate. However, when pressure exists, the lazy writer will continually be working to make room for new data coming into the buffer.

A certain indicator of memory pressure is ongoing lazy writes >0 with page reads/writes<0 and page life expectancy>600.

See also: 

SQL Server Memory Write latency by database file Avg ms/Write The average time in milliseconds each physical disk write is taking for a particular database file.

See "Database IO: Write latency" section for more details and performance ranges.

SQL Server Memory Checkpoint pages Avg/sec The average number of page writes each second to disk by the checkpoint process. Checkpoints flush all dirty buffer pages for a given database to disk and are a normal part of SQL Server operation. The frequency of checkpoints and volume of checkpoint pages is dictated directly by the "recovery interval" server option. SQL Server uses checkpoints to batch writes to disk, which is generally more efficient. However, if the volume of each checkpoint is too high and you see a correlation with high disk latency, it may indicate that the disk system isn't keeping up.

Checkpoints will sometimes correlate with lower page life expectancy, but only because when pages are flushed from memory, the disk system can drop them more efficiently later on. If you see correlation with high disk latency, it may indicate that the disk system isn't keeping up.

Checkpoints will sometimes correlate with lower page life expectancy, but only because when pages are flushed from memory, the disk system can drop them more efficiently later on. If you see correlation with high disk latency, it may indicate that the disk system isn't keeping up.

This can happen even when the disk is capable of easily sustaining higher throughput, and can lead to abnormal long checkpoints, and blocking of the log writer process.

DBCC TRACER (2002, 2004 – 1) will log many additional details about the checkpoint process to the SQL Server error log, and can be very helpful for troubleshooting checkpoint-related issues.

See also: 

SQL Server Memory Lazy writes Avg/sec The average number of writes per second by the lazy write. The lazy writer periodically scans the buffer and writes pages that have low use counts in order to maintain a certain number of pages on the free list. Ideally, this value should be at or near 0 most of the time. When there is no memory pressure, the lazy writer will generally issue data writes at an even rate. However, when pressure exists, the lazy writer will continually be working to make room for new data coming into the buffer.

A certain indicator of memory pressure is ongoing lazy writes >0 with page reads/writes<0 and page life expectancy>600.
**HYPER-V HOST METRICS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Performance Metric</th>
<th>Type</th>
<th>Description</th>
<th>Modes</th>
<th>Warning Range</th>
<th>Critical Range</th>
<th>May correlate with</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Usage (VM)</td>
<td>Usage time by Guest</td>
<td>%</td>
<td>The usage % time for each individual guest machine. Represented by the color selected for each individual Virtual Machine guest (consistent over the dashboard section).</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>CPU Usage (VM)</td>
<td>vCPU Wait time</td>
<td>%</td>
<td>The % time the guest virtual machine spent waiting for a Host kernel resource.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Hyper-V Migrating Memory (VM)</td>
<td>Memory by Virtual Machine Guest</td>
<td>MB</td>
<td>The amount of virtual memory used by each guest on the host machine.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Disk IO (VM)</td>
<td>IOPS (Read)</td>
<td>Reads/sec</td>
<td>The number of read operations per second across all Virtual Machines belonging to the Host. In Sample mode read operations are broken down by individual Virtual Machine. Use the drop-down to select an individual guest machine.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Disk IO (VM)</td>
<td>IOPS (Write)</td>
<td>Writes/sec</td>
<td>The number of write operations per second across all Virtual Machines belonging to the Host. In Sample mode write operations are broken down by individual Virtual Machine. Use the drop-down to select an individual guest machine.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Disk IO (VM)</td>
<td>Read Throughput</td>
<td>MB/sec</td>
<td>The average MB per second read across all Virtual Machines belonging to the Host. Use the drop-down to select an individual guest machine.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Disk IO (VM)</td>
<td>Write Throughput</td>
<td>MB/sec</td>
<td>The average MB per second written across all Virtual Machines belonging to the Host. Use the drop-down to select an individual guest machine.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
</tbody>
</table>

**AZURE SQL DATABASE METRICS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Performance Metric</th>
<th>Type</th>
<th>Description</th>
<th>Modes</th>
<th>Warning Range</th>
<th>Critical Range</th>
<th>May correlate with</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTU Usage</td>
<td>Total DTU %</td>
<td>%</td>
<td>A DTU represents the power of the database engine as a blended measure of CPU, memory, and read and write rates. This metric helps you assess the relative power of the SQL Database performance logs. Each Service Tier, which sets pricing and usage limits for an Azure SQL Database, expresses the amount of resource limits as a number of DTUs. The more DTUs an Azure SQL Database is allocated the more resources the database will have to service the workload. If your database is seeing high Total DTU % usage it may benefit from adjusting to the next higher Service Tier to improve performance. Likewise if you are consistently seeing very low Total DTU % usage you may save some money by scaling down to the lower Service Tier. For more information regarding DTUs read the documentation on Azure.com. For variable performing workloads and sharing resources across multiple Azure SQL Databases also see elastic Database Pools.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>DTU Usage</td>
<td>CPU %</td>
<td>%</td>
<td>This metric is the average CPU percentage based on the limit of the service tier. This is one of the metrics that makes up DTU.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>DTU Usage</td>
<td>Data I/O %</td>
<td>%</td>
<td>This metric is the average Data I/O percentage based on the limit of the service tier. This is one of the metrics that makes up DTU.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>DTU Usage</td>
<td>Log I/O %</td>
<td>%</td>
<td>This metric is the average Log I/O percentage based on the limit of the service tier. This is one of the metrics that makes up DTU.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Memory Usage</td>
<td>Allocated Memory Usage</td>
<td>%</td>
<td>Each Service Tier has a maximum amount of memory allowed for the Azure Database to use. This metric provides the percentage of the allowed memory being used for the database.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Database Size</td>
<td>Size in GB</td>
<td>%, GB</td>
<td>Each Service Tier has a maximum size allowed for the Azure SQL Database. This metric provides the size of the Database, not including the transaction log. If you are approaching the maximum size allowed you may need to increase your service tier size or look to scale out your database using clustering techniques. For more information about sharing with Azure SQL Database look at the elastic Database Pools.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>User Connections</td>
<td>Last value</td>
<td>The total number of connections established to the Azure SQL Database. Details for each connection can be viewed querying the sys.dm_exec_connections DMV.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Blocked Processes</td>
<td>Last value</td>
<td>The total number of blocked processes detected on the Azure SQL Database. The Dashboard is designed to give you ‘a glance’ information about the number of blocked processes.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Transactions</td>
<td>Avg/sec</td>
<td>The average number of transactions per second for the Azure SQL Database. A transaction can be either a user-defined statement block surrounded by BEGIN TRAN and END TRAN, or an individual DML statement (insert, update or delete).</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Backup MB/sec</td>
<td>MB/sec</td>
<td>The data rate in MB/sec for any backup operations taking place on the server. In Azure SQL Databases the platform creates backups automatically for you. While you do not have direct control over the backup times, this does allow you to help determine if backups are negatively affecting your performance.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>CPU % of Total Wats</td>
<td>%</td>
<td>CPU % of Total Wats represents the percentage of all work which see SQL Wait. Signal wait time is the time a thread has spent waiting on the CPU after being signaled that its resource is available. A High CPU % of Total Wats may indicate CPU pressure.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Wait Time By Category</td>
<td>Avg ms/sec</td>
<td>The average wait time (in milliseconds) per second for all processes on the database, broken down by major physical resource category (CPU, Memory, etc.) Although there are hundreds of wait types, only the relatively few wait types that can definitively be attributed to one of the physical resource categories are included in the calculations for this chart. The “Other” category is for a few important wait types that do not affect performance in one of more than one major category, or cannot be directly attributed to any category with absolute certainty, such as backups and parallelism respectively.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>SQL Server Activity</td>
<td>Wait Time By Class</td>
<td>Avg ms/sec</td>
<td>The average wait time (in milliseconds) per second for all processes on the server, broken down by wait class. There are approximately 30 different wait classes, and each represents a particular SQL Server function or area of type of activity. Meanings or incurred wait types such as timer and queue waits are pre-filtered from view. Each class is further broken down resource and CPU waits. If you hover over a class column, the detailed wait types are shown in a tooltip.</td>
<td>S, H</td>
<td>S</td>
<td>H</td>
<td></td>
</tr>
</tbody>
</table>
9.2 Processes

The Processes tab is available as part of Performance Analysis for Windows.

By default, Performance Analysis for Windows collects information about well-known or "categorized" processes. There are several of these predefined well-known process groups, including groups for SQL Server, IIS, SSAS, SSIS, and SSRS. If you would like to collect information about processes which are not in these default groups, you have a couple of different options for doing so:

- Add your own well-known groups and specify which processes belong to them.
- Configure an Uncategorized Process Filter, which allows for the collection of uncategorized processes

You may also choose to do both in tandem. For instance, you may wish to define several new well-known process groups which are pertinent to the monitored computer’s workload, so that information about those processes will always be collected. You may also wish to define an Uncategorized Process Filter, so that information about processes which are consuming a large amount of resources on the computer is also collected.

Jump to:

Adding new groups of well-known processes
Configure an Uncategorized Process Filter

### PROCESSES TAB DISPLAY

The Processes tab contains a grid view of all the processes about which you are collecting information. Processes are intelligently grouped by program and function, including groups for SSRS, SSIS, and SQL Server (well-known process groups). Metrics are displayed for each process, giving you insight into the resource usage for both the individual processes and their associated groups.

### OPTIONS

- **Show in groups** - This option groups well-known processes by their assigned group or category.
- **Showing/ HIDing Groups** - You can hide certain groups from view using the group drop-down listbox. Uncheck any group that you wish to hide.
- **Show well-known processes only** - This option will hide any processes which are not part of a well-known group.
- **Additional Columns** - Additional columns are available, and can be accessed by right-clicking on any

<table>
<thead>
<tr>
<th>Section</th>
<th>Performance Metric</th>
<th>Type</th>
<th>Description</th>
<th>Modes</th>
<th>May correlate with</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Active</td>
<td>%</td>
<td>Percentage of time that the virtual machine was ready, but could not get scheduled to run on the physical CPU. CPU ready time is dependent on the number of virtual machines on the host and their CPU load.</td>
<td>S, H</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Active</td>
<td>MB</td>
<td>Amount of memory that is actively used, as estimated by VMware based on recently touched memory pages.</td>
<td>S, H</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Consumed</td>
<td>MB</td>
<td>Amount of machine memory used on the host. Consumed memory includes any memory used by the Service Console, the Vmkernel, vSphere services, plus the total consumed metrics for all running virtual machines.</td>
<td>S, H</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Overhead</td>
<td>MB</td>
<td>Total of all overhead metrics (including VM overhead and Vmkernel overhead) for powered on virtual machines, plus the overhead of running vSphere services on the host.</td>
<td>S, H</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Granted</td>
<td>MB</td>
<td>Amount of machine memory or “physical” memory that is mapped for the host.</td>
<td>S, H</td>
<td></td>
</tr>
</tbody>
</table>
column header and selecting the Column Chooser option.

- **Sorting Columns** - You may sort the gridview by specific columns by selecting any column header. Additional sorting options are available from the context menu of any column header.

## ADDING NEW GROUPS OF WELL-KNOWN PROCESSES

By default, there are several well-known processes which are logically grouped by program/function, including predefined groups for SQL Server, IIS, SSAS, SSIS, and SSRS. You can add additional groups of your own using the following SentryOne Database tables:

- **PerformanceAnalysisDeviceProcessGroup**
- **PerformanceAnalysisDeviceProcessGroupMapping**

### PerformanceAnalysisDeviceProcessGroup

Use this table to define groups your Well-Known processes.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Identity column</td>
</tr>
<tr>
<td>ProcessGroupName</td>
<td>This is the group name which will be displayed in the Processes tab.</td>
</tr>
<tr>
<td>ProcessGroupNameFull</td>
<td>The full name is not displayed in the Processes tab.</td>
</tr>
<tr>
<td>ProcessGroupDescription</td>
<td>The description of the Process group.</td>
</tr>
</tbody>
</table>

### PerformanceAnalysisDeviceProcessGroupMapping

Use this table to map specific processes to the groups you defined in the PerformanceAnalysisDeviceProcessGroup table.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Identity column</td>
</tr>
<tr>
<td>PerformanceAnalysisDeviceProcessGroupID</td>
<td>The ID of the group you would like this process to map to. Should map to the ID column of a group found in the PerformanceAnalysisDeviceProcessGroup table.</td>
</tr>
<tr>
<td>ProcessName</td>
<td>Enter the exact name under which the process is executed, including the executable extension. For example: 'services.exe'</td>
</tr>
<tr>
<td>ServiceName</td>
<td>Enter the name of the service exactly as it is displayed on the General properties tab of the service as shown from the services applet.</td>
</tr>
<tr>
<td>ShowOnPADashboard</td>
<td>Specify if the group will be displayed on the Performance Analysis Dashboard.</td>
</tr>
<tr>
<td>CommandLineMatchRegex</td>
<td>This column is useful when differentiating processes which use the same executable, but perform different functions based upon the passed in parameters. The regex should use a valid .NET compatible regular expression as defined here. Note, that the match operation is case-insensitive.</td>
</tr>
</tbody>
</table>

### EXAMPLE

As an example, if the Target you are monitoring is a Microsoft Exchange Server, you may want to create a new group of well-known processes which includes those related services. In this example we will create a new group of well-known processes for **Microsoft Exchange**, and then we will add the following Microsoft Exchange related services to the group:

- Microsoft Exchange Information Store
- Microsoft Exchange Mailbox Assistants
- Microsoft Exchange Transport
Add the Processes and Map them to the Group

```
Insert INTO dbo.PerformanceAnalysisDeviceProcessGroupMapping
    (PerformanceAnalysisDeviceProcessGroupID, ProcessName, ServiceName, ShowOnPADashboard)
VALUES (9,'Store.exe','MSExchangeIS',1),
       (9,'MSExchangeMailboxAssistants.exe','MSExchangeMailboxAssistants',1),
       (9,'MSExchangeTransport.exe','MSExchangeTransport',1)
```

**Note:** Be sure that the `PerformanceAnalysisDeviceProcessGroupID` corresponds to the **ID column** of your desired group in the `PerformanceAnalysisDeviceProcessGroup` table.

### PROCESS COLLECTION SETTINGS

Process collection is controlled with the **Windows Instance Settings** and can be accessed as follows:

1. Select the desired **Windows Instance** in the **Navigator pane**
2. Open the **Settings pane (View menu -> Settings)**
3. In the **Settings pane** choose **Windows Instance** from the drop-down menu.

You should now see the Process Collection settings in the Settings pane. By default when a Windows Instance is monitored, process information will be collected about all **well-known** processes. You may disable the collection of process information by changing the **Collect Processes Setting** to False. You may enable the collection of additional processes (non well-known processes) by configuring an **Uncategorized Process Filter**.

### CONFIGURE AN UNCATEGORIZED PROCESS FILTER

By default, Performance Analysis for Windows collects information about "Well-Known" or "Categorized" processes. You may configure **SentryOne** to also collect information about Uncategorized processes by specifying an Uncategorized Process Filter. This filter may be built around various metrics, including % CPU time or Read and Write bytes per second.

Follow the directions in the previous section to access the Process Collection Settings for your Windows Instance. Next, select the Uncategorized Process Filter row and use the ellipsis (…) button to open the Filter Editor. Add rules to the filter as desired. The Filter Editor was designed with flexibility in mind, and allows you to specify any number of criteria around events. You can define a complex rule with multiple groups and logical operators. For more information about using the Filter Editor see the **Filter Editor topic**.

---

**9.3** Disk Activity

**Applies to:** SQL Sentry, Win Sentry, BI Sentry, and V Sentry

Disk bottlenecks represent one of the most common sources of performance problems for Windows and SQL Server. The **Disk Activity** tab provides a patented graphical disk analysis system which breaks down disk activity and latency at the controller, physical disk, and file level, highlighting bottlenecks at any point in a disk system.

The top pane of the display shows a graphical representation of the entire disk system as Windows sees it. The bottom pane displays **activity metrics** in either a hierarchical disk system format, or database list format.
DISK ACTIVITY MODES

The Disk Activity tab has two modes, Real Time and History. The active mode is controlled with the auto-refresh toolbar button. When the auto-refresh Play button is selected, the tab enters Real Time mode. Data for the last collected sample is shown, and moving segmented lines indicate the I/O direction (read or write) and latency for each disk, file, and controller.

- The top pipe represents read activity
- The bottom pipe represents write activity

The thickness and color of the displayed segmented line indicates the amount of latency being experienced. As latency increases, the lines grow in width and change through green, olive, orange, and red to give a quick visual indication of the decreased performance.

Selecting any file will display a tooltip which shows the numerical values for the reads and writes hitting a particular file. The amount of overall latency being experienced by the disk is distinguished from the amount of latency for individual files. The overall disk latency is represented with the horizontal pipe connecting all of the individual files, while the small vertical pipes represent latency at the file level.

The table below lists the color of the segmented line and the corresponding range of latency values. All latency values are listed in milliseconds (ms).

<table>
<thead>
<tr>
<th>Color Range</th>
<th>Read Latency</th>
<th>Write Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>&lt; 10 ms</td>
<td>&lt; 10 ms</td>
</tr>
<tr>
<td>Greenish-Yellow to Orangish-Yellow</td>
<td>10-20 ms</td>
<td>10-20 ms</td>
</tr>
<tr>
<td>Orange to Reddish-Orange</td>
<td>21-30 ms</td>
<td>21-40 ms</td>
</tr>
<tr>
<td>Red</td>
<td>&gt; 30 ms</td>
<td>&gt; 40 ms</td>
</tr>
</tbody>
</table>

Read and Write Latency color gradients

When the auto-refresh Pause button is selected, the tab goes into History mode. Aggregate disk metrics are
displayed for the selected date range, and the solid lines are used to indicate latency. History mode enables you to see how your disk system was performing over any time period.

**TOOLBAR**
- Disk Grid View / Database Grid View - When auto-refresh is paused, these viewing options are available.
- Show Empty Disks - Controls whether or not disks without any SQL Server database files are displayed.
- Order By Activity - Sort physical disks by activity in descending order.
- Sum / Average - When auto-refresh is paused, these viewing options are available.

**CONTEXT MENUS**
- Left click - Sets the focus to any file. Other files for the same database are also highlighted, and the corresponding grid row.
- Right click - Run a Quick Trace or a Quick Report.

**EXPORTING AND REPORTING OPTIONS**

**Exporting Data** - The Disk Activity grid view can be exported through the file menu (File → Export Data).

**Quick Report** - If you need access to database level I/O metrics in a report format, a quick report can be run from the right-click context menu of any file. You may run a Quick Report while in Real Time mode for the last 10 minutes of data.

**DISK ACTIVITY METRICS**

The following table lists descriptions of the metrics displayed on the Disk Activity screen. For more information on interpreting latency values be sure to see Disk I/O: Read latency by physical disk.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Machine</td>
<td>The Virtual Machine that the file is associated with.</td>
</tr>
<tr>
<td>NOTE:</td>
<td>This column will only appear for targets that are also Hyper-V Hosts, and the column will only be populated for files directly associated with Virtual Machines on the host (e.g., .vhdx files, mounted .iso files).</td>
</tr>
<tr>
<td>Read Bytes</td>
<td>The average number of bytes read per second over the last polling interval.</td>
</tr>
<tr>
<td>History</td>
<td><strong>Average Mode</strong> - The average number of bytes read per second over the selected range.</td>
</tr>
<tr>
<td></td>
<td><strong>Sum Mode</strong> - The total number of bytes read for the selected range</td>
</tr>
<tr>
<td></td>
<td><strong>Real Time</strong> - The percentage of bytes read from this file in relation to other files belonging to the Target, over the last sample polling interval.</td>
</tr>
</tbody>
</table>

The sample collection frequency for disk counters is 20 seconds; while in Real Time mode the Disk Activity screen will update according to this sample interval. A custom sample interval can be specified. This may be useful if you are monitoring a server with a large number of database files per database, as this can lead to increased SentryOne repository growth and transaction log activity. For more information about adjusting the disk counter collection frequency see this KB article.
### Name

**Average Mode** - The average percentage of bytes read from this file, in relation to other files belonging to the Target, over the selected historical range.

**Sum Mode** - n/a

**Real Time**
The number of read operations per second, over the last sample polling interval.

**History**

**Average Mode** - The average number of read operations per second over the selected range.

**Sum Mode** - The total number read operations occurring in the selected range.

**ms/Read**
The average time in milliseconds each physical disk read took over the last sample polling interval.

**History**

**Average Mode** - The average time in milliseconds each physical disk read took over the selected time range.

**Sum Mode** - n/a

**Real Time**
The average number of bytes written per second over the last polling sample interval.

**History**

**Average Mode** - The average number of bytes written per second over the selected range.

**Sum Mode** - The total number of bytes written for the selected range

**% Write Bytes**
The percentage of bytes written to this file in relation to other files belonging to the Target, over the last sample polling interval.

**History**

**Average Mode** - The average percentage of bytes written to this file, in relation to other files belonging to the Target, over the selected historical range.

**Sum Mode** - n/a

**Real Time**
The number of write operations per second, over the last sample polling interval.

**History**

**Average Mode** - The average number of write operations per second over the selected range.

**Sum Mode** - The total number write operations occurring in the selected range.

**Real Time**

Milliseconds per write. The average time in milliseconds each physical write took over the last sample polling interval.
### Disk Space

The **Disk Space** tab contains information about disk space utilization within your environment, broken down by database and file. Use the **Disk Space** tab to identify disk capacity issues, understand where various database files reside on the disk system, and to help you in determining whether available disk space is being used optimally.

The **Disk space** tab has two distinct areas. The top pane is a graphical representation of the database layout at the disk level. Data files are represented with a unique solid color, and each associated log file is represented with the same color with a line overlay. This color is repeated for each file in the gridview found in the bottom pane. Selecting a file in either pane will highlight the associated representation in the other pane.

#### DISK SPACE MODES

The **Disk Space** tab has two modes, **Real Time** and **History**. The active mode is controlled by the auto-refresh toolbar button. When the auto-refresh **Play** button is selected, the tab enters **Real**
**Time** mode. Data for the last collected sample is shown.

When the auto-refresh **Pause** button is selected, the tab enters **History** mode. History mode displays the average disk space information over the specified time range. You may change the time range you are viewing by using the **Start** and **End** times and selecting the **Go** button in the toolbar.

**DISK SPACE USAGE BAND**

When index collection is enabled, index information is viewable by drilling in to the database level in the grid view. When a database is selected in the grid, a band displaying the space usage of that database's indexes will appear between the disk layout and the grid. Each box represents an index within the selected database. The size of each box illustrates the total amount of space used in relation to the other indexes within that database.

**TOOLBAR**

Once the **Disk Space** tab is in history mode the **Disk/Database Grid view** button becomes available in the toolbar. Selecting Database Grid View removes the groupings, and can be helpful when you are interested in viewing and sorting data across the entire disk subsystem.

- Disk Grid View / Database Grid View - When auto-refresh is paused, these viewing options are available.
- Show Empty Disks - Controls whether or not disks without any SQL Server database files are displayed.
- Sum / Average - When auto-refresh is paused, these viewing options are available.

**ADDITIONAL OPTIONS**

- Click on any row to set focus to the associated database and log file.
- Right click - Copy cell/row/all options available.
- Click the + command of any data file to show table and index information. For additional information about those metrics be sure to see the **Indexes topic**.

**EXPORTING AND REPORTING OPTIONS**

**Exporting Data** - The Disk Space grid view can be exported through the file menu (**File → Export Data**).

**Quick Report** - If you need access to database level size metrics in a report format, a quick report can be run from the right-click context menu of any file. You may run a Quick Report while in Real Time mode for the last 7 days of data.

**Disk/File Space Reports** - The **Disk / File Space Reports** can provide disk and database file level metrics. Access them through the Reports menu (**Reports → Performance Analysis → Disk/File Space**).

**DISK SPACE METRICS**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Machine</td>
<td>The Virtual Machine that the file is associated with.</td>
</tr>
<tr>
<td>SQL Server</td>
<td>The file's associated SQL Server.</td>
</tr>
<tr>
<td>Database</td>
<td>The file's associated database.</td>
</tr>
<tr>
<td>File</td>
<td>The name of the file.</td>
</tr>
<tr>
<td>Filegroup</td>
<td>The filegroup of the associated file.</td>
</tr>
</tbody>
</table>

**NOTE:** This column will only appear for targets that are also Hyper-V Hosts, and the column will only be populated for files directly associated with Virtual Machines on the Host (e.g., .vhdx files, mounted .iso files).

For general information about files and filegroups within SQL Server see this MSDN article: [Files and Filegroups Architecture](https://docs.microsoft.com/en-us/sql/relational-databases/tables/understanding-pages-and-extents)

**Type**
Specifies the type of the file, being either data or transaction log.

**Size (MB)**
The size in (MB) of the selected file.

**Used (MB)**
The amount of the disk space which has been allocated for the file and is in use.

**Used %**
The percentage of disk space which has been allocated for the file and is in use.

**Warnings**
- A pink highlight indicates that the autogrowth amount is greater than the remaining free space.
- A yellow highlight indicates that autogrowth is set as a percent.
- A tan highlight indicates that autogrowth is less than 10 MB or greater than 1 GB.

**Max File Size (MB)**
The maximum file size allowed for this data or log file.

The total number of virtual log files (VLFs) which compose the transaction log. Each physical transaction log is made up of a number of smaller VLFs. When the transaction log grows, by either a manual action, or because of an autogrowth, the number of virtual log files which compose the transaction log will also increase.

The actual number of corresponding VLFs which are added during any log growth is dependent upon the size of the growth. Reference the below table:

<table>
<thead>
<tr>
<th>Autogrowth size</th>
<th>VLFs created</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 64 MB</td>
<td>4</td>
</tr>
</tbody>
</table>
If the transaction log has been allowed to grow frequently in small increments, you may see a large number of VLFs. A log file containing a large number of VLFs may decrease database performance.

For general information about transaction log architecture see this MSDN article.

**Warnings**

- A tan highlight indicates that Total VLFs are greater than 100.
- A pink highlight indicates that Total VLFs are greater than 300.

The number of virtual log files (VLFs) containing log records which are still needed. There are a number of criteria for defining when a log record is needed; for more information see this Technet article. A log record which is still needed is defined as an active log record, and it follows that any VLF containing at least on active log record is an Active VLF.

### Active VLFs

- **Min VLF Size (MB)**: The size (MB) of the smallest existing VLF.
- **Max VLF Size (MB)**: The size (MB) of the largest existing VLF.
- **Avg VLF Size (MB)**: The average size (MB) of those VLFs which compose the log.

The number of VLFs which will be created during the next AutoGrowth event. When the transaction log grows, by either a manual action, or because of an autogrowth, the number of virtual log files which compose the transaction log will also increase. This number is dependent upon the size of the growth, reference the table below.

### Autogrowth VLF count

<table>
<thead>
<tr>
<th>Autogrowth size</th>
<th>VLFs created</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 64 MB</td>
<td>4</td>
</tr>
<tr>
<td>&gt;= 64 MB and &lt; 1 GB</td>
<td>8</td>
</tr>
<tr>
<td>&gt;= 1 GB</td>
<td>16</td>
</tr>
</tbody>
</table>

The size (MB) of each VLF which will be created during the next autogrowth event, given the current Autogrowth settings.

### Autogrowth VLF Size (MB)

The last recorded backup time for the associated file.

### Last Backup Time

- A pink highlight indicates that there are no recent transaction log backups.

### Last Backup Type

The last recorded backup type for the associated file.

### Log Reuse Wait

This indicates what the transaction log is waiting on, in regards to reusing its space.

### File Path

The full file path of the data or log file.
The Top SQL tab lists all TSQL batches, stored procedures (RPCs), and statements events collected for the current date range that exceeded the Minimum Duration specified in the Settings pane. The default filter is Duration, but filtering using CPU and IO is also available. The default minimum duration is 5 seconds, but this can be adjusted up or down as needed through the Top SQL Source Settings.

In order to collect Top SQL metrics for Azure SQL Database Targets the “Allow SentryOne Monitoring Objects in Target” setting under the Azure SQL Database Connection settings category must also be set to True. This option will create a few objects within a SentryOne schema on the target database.

**HOW CAN I CHANGE WHAT IS COLLECTED FOR TOP SQL?**

You can change the Top SQL which is collected by SentryOne through the Top SQL Source Settings. You can change the Top SQL events collected by SentryOne through the Top SQL Source Settings at these levels: All Targets (Global), Site, Target Group, Target, and Instance.

More Information and Examples of Changing Top SQL Settings

See below for examples:

If you wanted to configure the Top SQL Minimum Duration Collection Setting Globally:

1. Select the Shared Groups node in the Navigator Pane.
2. In the Settings pane, use the top drop-down list and select SQL Server Settings. If you do not see the Settings pane once you have selected your desired node in the Navigator pane, use the View Menu (View → Settings).
3. Use the second drop-down list to select Top SQL Source. You should now see the Top SQL Source Settings that are being applied Globally.
4. Change the Minimum Duration to the desired value, it will be saved automatically.

If you wanted to configure the Top SQL Minimum Duration Collection Setting for an individual Instance:

1. Select the desired Instance node in the Navigator pane.
2. In the Settings pane, use the top drop-down list and select SQL Server Settings. If you do not see the Settings pane once you have selected your desired node in the Navigator pane, use the View Menu (View → Settings).
3. Use the second drop-down list to select Top SQL Source. You should now see the Top SQL Source Settings that are configured for the Instance.
4. Within the Collection Settings Section change the Inherit From Parent Setting to False.
5. Change the Minimum Duration to the desired value, it will be saved automatically.

There are a few additional things you should know regarding the Top SQL Source Settings.

There is an AND relationship that exists between the Minimum Duration, Minimum CPU, Minimum Reads, and the Minimum Writes Collection Settings. This means, that in order to be collected as Top SQL events, the duration and the amount of CPU and IO must exceed the specified thresholds.
SQL, the event will need to satisfy each individual Collection Setting. For example, if you set the Minimum Duration at 10 seconds and the Minimum Reads at 25, an event would need to meet both a Minimum Duration of 10 seconds AND a Minimum Reads of 25 to be captured in Top SQL.

Minimum Duration cannot be set below 100ms unless Minimum CPU, Minimum Reads, or Minimum Writes is greater than 0. This lower limit is enforced because setting this thresholds below 100ms for an extended period of time could dramatically increase the volume of data collected and stored by SentryOne, and have a negative impact on the monitored server. SentryOne's Quick Trace functionality is better suited to analyze extremely short duration events.

**HOW CAN I ADJUST CAPTURED TEXTDATA LENGTH?**

By default textdata for captured Top SQL events is truncated at 10,000 characters. You can change this maximum length in the Monitoring Service Settings (Navigator → Configuration → Global Settings → Advanced → Query Collection Max Text Length).

**HOW CAN I HIDE QUERIES WITHIN TOP SQL AND STOP ALERTS ABOUT THEM?**

You can hide individual queries or groups of queries with the right-click context menu Hide command.

- To hide a single query simply select the row you wish to hide, right-click and choose the Hide command.
- To hide a group of queries, first drag a column header to the top of the grid view to group like items. Next select the group header in the grid view and use the right-click context menu Hide command.

Once a query has been hidden in Top SQL you will no longer be alerted about it for any of the Top SQL related Conditions. You can unhide all hidden queries by selecting the Show Hidden button from the toolbar. To unhide an individual query permanently use the Visible (available in the column chooser) - checkbox that toggles the effect of "Hide/Hide Group".

**DISPLAY PANES**

The Filter pane provides the following filters for controlling which records will be displayed: SQL Servers, Applications, Databases, and Hosts.

Select any combination of items to set the filters. To select more than one item, use shift-click or ctrl-click. To view the filtered results click the Refresh toolbar button or F5.

Show Top Control- Depending on the length of the displayed interval, you may have a lot of data to work with, use this setting to control how many rows are retrieved.

> The Filter pane is not displayed on an Azure SQL Database target.

The Running Queries pane shows all running queries matching the current filters and is independent of the time range setting. The View button in the Plan column will open a Plan Explorer session for the associated query. The Text Data column shows the original text the client sent to the server and the Active Statement column shows the currently running statement.

Running Queries is not supported for SQL Server 2000 or Azure SQL Database.

> Note: The text data for the active statement is displayed in the Text Data pane when the Active Statement cell is selected.

The Completed Queries pane lists all records matching the current filters. The View
button in the Plan column will open a Plan Explorer session for the associated query. The
default filter is to display all records. There are two modes for this pane: Default and
Totals.

Default mode lists presents all events for the active date range in a standard list format,
sorted descending by End Time by default. Default mode is activated whenever Auto-
refresh is enabled by clicking the “Play” button on the toolbar, in which case the events for
the last 10 minutes will be shown, with new events coming in automatically at the top of
the list.

Totals mode is enabled by selecting the Show Totals button. This will group all like events
 together using a normalized version of the TSQL with all variable parameters replaced, and
will show aggregates for each group. This enables you to quickly determine which events
are responsible for the most CPU or IO activity for the active date range.

SentryOne 8.2 introduces enhanced SQL text data normalization. If upgrading from a build prior to 8.2, enhanced normalization
must be activated in the Monitoring Service Settings > Performance Monitor tab.

The Procedure Stats pane provides aggregate information for all procedures matching the
current filters. The View Plan button in the Plan column will open a Plan Explorer session
for the associated procedure. Additional historical information may be available in the
Runtime Stats. You can reach Runtime Stats by right clicking on a procedure and selecting
Jump to -> Runtime Stats.

SentryOne installations upgraded from a version older than 8.2 will need to enable the Collect Query Stats option in the Top SQL
Source setting.

Procedure Stats is not available for SQL Server 2000 or SQL Server 2005.

The Query Stats pane provides aggregate information for queries matching the current
filter. This area will also provide insight into queries that run very quickly and in high
volume.

SentryOne installations upgraded from a version older than 8.2 will need to enable the Collect Query Stats option in the Top SQL
Source setting.

Query Stats is not available for SQL Server 2000 or SQL Server 2005.

The Text Data pane shows the TSQL text data for the currently highlighted Top SQL record with syntax highlighting.

PROCEDURE STATS AND QUERY STATS COLLECTION

There are 3 settings that can be adjusted to change how long a query must run to be collected. These
settings are located in the Top SQL Source area in the Settings pane.

- Query Stats Sample Interval – Specifies how often to sample query stats
- Filter Time Span – Specifies the base length of time over which the collection filters will be
  applied to Query Stats
- Minimum Duration – Specifies the minimum duration for a statement to be logged
- Filter Factor – The Filter Factor is calculated by dividing the Query Stats Sample Interval by the
  Filter Time Span. The collection filters such as Minimum Duration are multiplied by this value
  when applied to Query Stats collection.

Each of these Settings play a part in what is effectively captured for Query Stats in your environment. As
an example, using the following default settings:

- Query Stats Sample Interval = 60 seconds
- Filter Time Span = 30 seconds
Minimum Duration = 5 seconds

The total cumulative duration of all executions of a query over the 60 second sample interval must be greater than 10 seconds.

\[
\frac{\text{Query Stats Sample Interval (60 sec)}}{\text{Filter Time Span (30 sec)}} = \text{Filter Factor (2)}
\]

Minimum Duration (5 sec) \times Filter Factor (2) = Total cumulative duration threshold (10 sec)

**CONTROLS**

**TOP PANES**

- Filter (minimized by default) - After changing any of the filters, it is necessary to click the Refresh button, or press F5, to apply the filter.
- Click on a combination of items in any of the lists to set the filters. The default filter is to display all records.
- To select more than one item in a list, use shift-click or ctrl-click.
- Minimum Duration - set to filter by the minimum run time for Top SQL events.
- Show Top - Depending on the length of the displayed interval, you may have a lot of data to work with. This setting controls how many of the rows are actually retrieved.

**ADDITIONAL OPTIONS**

- Grouped (main toolbar) - this is an on or off option. When off, every Top SQL event is listed. When on, the Text Data behind the event is normalized and then the list is grouped by that normalized data.
- Show Hidden (main toolbar) - You can hide individual lines or groups of rows. This will toggle the display of those lines. See “Visible” below.
- Context Items
  - Right click on a Top SQL event
  - Hide/Hide Group - hides the selected Top SQL events.

**NOTE:** This will also disable any alerts for the events, and prevent them from being displayed on the Event Calendar

- Jump to Dashboard - when Show Totals mode is enabled all events belonging to the selected group will be overlaid on the Dashboard. When the Default mode is enabled only the selected event will be overlaid on the Dashboard.
- Jump to Calendar - go to that event on the Event Calendar (requires server to be watched with EM).
- Jump to Runtime Stats - opens a tab showing the runtime stats for the event.
- Jump to Plan Explorer - opens the Plan Explorer tab in the client.
- Kill Process (running queries pane only) - option to kill the running process.

- Visible (available in the column chooser) - checkbox that toggles the effect of “Hide/Hide Group.”

**TOP SQL METRICS**

**TOP SQL Metric**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>The SQL Server instance where the associated event took place</td>
</tr>
<tr>
<td>Plan</td>
<td>Use the View button to open a new Plan Explorer Session for the captured plan.</td>
</tr>
<tr>
<td>Text Data</td>
<td>Associated text data captured for the event.</td>
</tr>
<tr>
<td>Event Class</td>
<td>The event class of the associated event, see also: <a href="#">MSDN Event Class Reference</a></td>
</tr>
<tr>
<td>Host</td>
<td>Name of the Target on which the associated event is running.</td>
</tr>
<tr>
<td>Login</td>
<td>The Windows or SQL Server account associated with the event.</td>
</tr>
<tr>
<td>Application</td>
<td>Name of the client application which created the connection.</td>
</tr>
<tr>
<td>Database</td>
<td>Name of the database for the associated event.</td>
</tr>
<tr>
<td>Duration</td>
<td>The amount of time taken by the associated event.</td>
</tr>
<tr>
<td>CPU</td>
<td>The amount of CPU time (in milliseconds) used by the event.</td>
</tr>
<tr>
<td>CPU %</td>
<td>The percentage of CPU time used by this event, in relation to any other Top SQL Events which are shown in the grid.</td>
</tr>
<tr>
<td>Reads</td>
<td>The number of logical disk reads performed by the server on behalf the event.</td>
</tr>
<tr>
<td>Reads %</td>
<td>The percentage of bytes read by this event, in relation to any other Events which are shown in the Top SQL grid.</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of physical disk writes performed by the server on behalf of the event.</td>
</tr>
<tr>
<td>Writes %</td>
<td>The percentage of bytes written by this event, in relation to any other Events which are shown in the Top SQL grid.</td>
</tr>
<tr>
<td>Start Time</td>
<td>Time the event started.</td>
</tr>
<tr>
<td>End Time</td>
<td>Time the event ended.</td>
</tr>
<tr>
<td>Information</td>
<td>Additional data captured about the event.</td>
</tr>
<tr>
<td>Error</td>
<td>Error number of a given event.</td>
</tr>
<tr>
<td>SPID</td>
<td>The server process ID (SPID) that is assigned to the process.</td>
</tr>
<tr>
<td>Host Process ID</td>
<td>The ID assigned by the host computer to the process where the client application is running.</td>
</tr>
<tr>
<td>Has Plan</td>
<td>True if the associated event has a captured plan</td>
</tr>
<tr>
<td>Has Statements</td>
<td>True if the associated event has captured statements</td>
</tr>
<tr>
<td>Visible</td>
<td>The Visible column specifies the default visibility for the select row. Queries may be hidden in Top SQL.</td>
</tr>
</tbody>
</table>

### 9.5.1 Plan Explorer (integrated)

Plan Explorer is designed to make query plan analysis fast and intuitive. There are a number of ways to start a new Plan Explorer session within SentryOne. On the Top SQL tab, the View button in the Plan column will open a Plan Explorer session for the associated completed or running query. You may also start a new Plan Explorer session with the New Plan Explorer Session toolbar button or through the File menu.

**Plan Explorer Sessions** (.pesession) are designed to help you manage an historical record as you fine tune queries. By default, an historical entry will be generated as part of the Plan Explorer Session during estimated and actual plan retrieval. See also: [Plan Explorer Sessions](#)

💡 SQL Server MVP Aaron Bertrand has written extensively about the available features and benefits of using SentryOne Plan Explorer. You can find all of his Plan Explorer related posts at this link.

**GENERAL**
The screen and window layout of Plan Explorer is persisted by default, and can be saved to and loaded from disk. Default horizontal and vertical layout styles are included; the horizontal layout is better suited for shorter queries with wide plans, while the vertical layout is better suited for longer queries. The layout can be changed by selecting the desired layout button on the toolbar or through the context menu of any section’s header.

Entire plans can be saved to and loaded from file, complete with all statement and plan information. You can save a plan with the **Save button** located on the toolbar or through the **File** menu. A **Save As** command is also available from the File menu, allowing you to keep different versions of similar plans.

**Customizable Tab Layout**

The Plan Explorer tab layout can be completely customized. Each tab can be fully undocked from its parent window, allowing you to simultaneously view the plan diagram alongside any other chosen query detail tabs. To undock a tab from its parent window, simply click and drag the tab’s title bar. You may choose to either re-dock the tab with an existing tab, or float the tab in a standalone window. Use the push pin to unpin the tab from the immediate docked view, and it will be auto-hidden when not in use.

**STATEMENTS SECTION**

The **Statements Tree** tab shows a tree list representation of the entire query call stack, including all statements, conditional logic structures, looping structures, nested procedure calls, and dynamic SQL calls. Estimated costs (Total, CPU, and IO) and estimated rows are displayed for each statement, and actual costs and rows are shown side-by-side for any statements captured via Top SQL collection. Any significant differences between estimated and actual rows for captured statements are highlighted, making it easy to spot cases where statistics may be stale.

Each of the three sections will be automatically synchronized when you select a statement. This includes all the tabs in the Plan Details section. Use the Statements Tree to quickly find the highest cost statements. Selecting any statement will show the graphical diagram, top operations list, columns list, and tree list view for the statement’s execution plan.

The **Command Text** tab allows editing of the SQL for estimated and actual plan retrieval.

**NOTE:** When using a column filter that uses a percent value, use a decimal value for the filter. For example, if you’d like to the filter to be 50%, use 0.5.

**QUERY INFORMATION SECTION**

The **Text Data** tab displays the color-coded TSQL batch or procedure definition of the plan. You may save the text with the **Save SQL** context menu command or copy the the text data to the Command Text tab with the **Copy To Command Text** context menu command. The **Plan XML** tab displays the execution plan in XML format. You may save the plan in .sqlplan format, with the **Save Execution Plan** context menu command.

**PLAN DETAILS SECTION**

When you select a statement in the Statements Section all the tabs in the **Plan Details** section will be synchronized with the selected statement. To inspect or make modifications to an index, the index properties window can be opened by right-clicking any index operation node in the **Plan Diagram** tab, or any grid row within the other Plan Details tabs. Likewise you can create new indexes via a separate context item. When you select an operation in any of the Plan Details tabs it will be automatically synchronized. You may use the **Copy** command, available through the right-
click context menu, to copy any of the grid view data in the Plan Details section.

The **Plan Diagram** tab has an Optimized layout algorithm which renders plans in a much more condensed view than SSMS, so more of the plan fits on the screen without having to zoom out. If needed, you can easily zoom in and out by holding the ctrl key and using your mouse scroll wheel. Optimized plan node labels prevent truncation of object names in most cases. To disable truncation completely so full object names are always visible use the **Full Object Names** context menu command.

The estimated cost of the operation is displayed above the each node for maximum readability. These cost labels use color scaling by CPU, IO, or CPU+IO so highest cost operations are instantly obvious, even on larger plans. CPU+IO is used by default, this can be changed through the context menu **Costs By** option. All costs in the Plan diagram are shown to the first decimal place.

Through the context menu of the Plan Diagram you may also choose to show cumulative costs in lieu of per node costs; when combined with color scaling this feature makes it easy to see which subtrees are contributing most to the plan cost.

The connector line width can be scaled by either row or data size with the **Line Widths By** context menu command. The metric you choose to scale by, will be displayed above all connector lines within the Plan Diagram.

The **Plan Tree** tab is a tree representation of the plan which shows all operations and associated metrics. Use the arrows to expand and collapse sections of the Plan Tree. Right click any column header and use the Column Chooser command to access additional plan metrics. Any significant differences between estimates and actuals are highlighted as are possible problematic operations like scans and bookmark lookups.

The **Top Operations** tab contains a sortable list view of all plan operations. The list view is sorted by total cost descending by default so you can immediately see which operations are the highest cost.

The **Query Columns** tab shows exactly how data is accessed for each column used by the query, including the associated operation and estimated rows. This view makes it easy to identify bookmark lookups or index scans occurring because indexes are non-covering.

The **Parameter** tab shows all statement parameters with runtime and compiled values.

When an Actual Plan is captured, or generated the **Table I/O** tab will be shown. This tab breaks down all the reads from a query by object, making it very easy to see where the majority of your I/O is coming from.

The **Wait Stats** tab was introduced as part of the feature set of SentryOne 7.2. When an Actual Plan is retrieved with related wait stats, the Wait Stats tab will be displayed next to the Results tab. The Wait Type is shown along with any associated Wait Time or Signal Time. The Wait Stats tab can help you in identifying potential resource bottlenecks, including those related to memory pressure, CPU pressure, disk I/O, and networking.

**Note:** Wait stats are only available for queries executed against servers running SQL Server 2008 or newer.

### PLAN HISTORY

SentryOne tracks all plan versions for a particular query over time, so you can easily determine when a plan has changed and caused query performance problems. The plans are the estimated
9.5.1.1 Plan Explorer Sessions

**Plan Explorer Sessions** are designed to help you manage an historical record as you fine tune queries. By default, an historical entry will be generated as part of the Plan Explorer Session during estimated and actual plan retrieval.

*Note:* You can change the behavior of how Plan Explorer generates new historical entries through the User Preferences (Tools → User Preferences → Performance Analysis tab). The Only save history when command text or Instance settings change option controls this behavior.

**VERSION HISTORY**

Each historical entry retains all captured plan details and metrics within the various Plan Explorer tabs. Each version is associated with a unique version number. **Plan Explorer** includes a Plan History pane (View menu → Plan History) which allows you to quickly navigate through the different versions within the active Plan Explorer Session. You can also delete a version with the Delete command accessed through the Plan History pane context menu.

**ADDING COMMENTS**

**Plan Explorer** also allows you to add comments to each historical version (View menu → Comments). This allows you to easily keep track of the reasoning behind any changes you make, such as changes to the command text or any indexing optimization.

**MULTIPLE SESSIONS AND TABS**

As each session is managed within its own tab, multiple **Plan Explorer Sessions** can be open at the same time. SentryOne was designed with a multiple document interface allowing you a number of options when managing tabbed sessions, including the ability to arrange tabs both horizontally and vertically, and the ability to tear off tabs. Tab windowing options can be found in the Window menu.

**SHARING SESSION FILES**

**Plan Explorer Session files** can be easily shared with others, even if they don't have access to the full SentryOne Client.

**SentryOne Plan Explorer** is a stand-alone query analysis tool. **Plan Explorer** uses session files in managing history, just like the integrated Plan Explorer, and contains complete support for opening Plan Explorer Session files generated in the SentryOne Client.

**Plan Explorer** can also open session files in a limited fashion. When you open a **Plan Explorer Session file** (.pesession) in **Plan Explorer** you are presented with a list of all the historical entries in that session. The list contains information about each historical entry, including version numbers and comments previously added. Once a user selects a version they wish to open, it will be opened as a .queryanalysis file.

This is possible because each historical entry in a **Plan Explorer Session** is actually a .queryanalysis file. Plan Explorer Session files are archive files, containing both the individual .queryanalysis files...
and metadata about the session.

You can also save any single historical entry of a Plan Explorer Session as a stand-alone .queryanalysis file. With the entry you wish to save active, use the Save As command (File menu → Save as) and choose the desired file type.

### STARTING A NEW SESSION

You can start a new Plan Explorer Session a number of ways, including:

- File menu → New
- From the Toolbar use the New Plan Explorer Session button
- Dragging and dropping a plan file onto the application

### SAVING A SESSION

You can save a Plan Explorer Session as follows:

- File menu → Save / Save As
- From the Toolbar use the Save button

---

**9.5.1.2 SQLPerformance.com Plan Upload**

SentryOne Plan Explorer includes support for uploading several different plan file types directly to SQLPerformance.com from the client. This feature allows you to post a question, with an included plan, to the SQLPerformance.com Q&A site answers.sqlperformance.com.

**Note:** For further information about SQLPerformance.com and the Plan Explorer Plan Upload feature please see this announcement by SQL Server MVP Aaron Bertrand.

You may post the following file types directly to SQLPerformance.com from the Plan Explorer Client:

- .pesession
- .queryanalysis
- .sqlplan *
- .xml *

**Note:** When you post either a .sqlplan file or a .xml file through the SentryOne Client to SQLPerformance.com it will automatically be converted to a .queryanalysis file.

### POSTING A PLAN TO SQLPERFORMANCE.COM

There are two different options for posting plans to SQLPerformance.com from the SentryOne Client. The first time you post a plan to SQLPerformance.com you will be prompted to register. Your login credentials for SQLPerformance.com will be saved locally in a secure manner. Once you have uploaded a plan to SQLPerformance.com a web link is embedded in the file. This link will allow you to jump directly to your post. This link will also be available for anyone who downloads the plan.
From the context of any open plan within Plan Explorer, you may use the Post To SQLPerformance.com button found on the toolbar. This will open a question form, automatically attaching the current plan, allowing you to add your desired comment or question.

You also have the option to Anonymize a query plan before uploading it by using the associated check box. See also: Anonymize Plans.

Alternatively you may upload a plan from the Help Menu (Help → SQLPerformance.com Post). Using this method you are not restricted to the currently open plan. You may upload any plan of your choice.

Note: When you upload a plan from the Help Menu a link back to the post will not be embedded in the local file. The embedded link will be available for those who download the plan from SQLPerformance.com.

POSTING .SQLPLAN AND .XML FILES

In addition to SentryOne Plan Explorer’s file types (.queryanalysis, .pesession), you may also post .sqlplan and .xml file types directly to SQLPerformance.com through the Plan Explorer Client. During the upload process these files will be automatically converted to the .queryanalysis file type.

SECURITY OBFUSCATION

Regardless of how you upload a plan, connection parameters stored by SentryOne Plan Explorer are removed from the plan during the upload process.

The following identifying connection parameters are automatically removed from plans during the upload process:

- Instance
- Database
- Login

ANONYMIZE PLANS

SentryOne Plan Explorer has the ability to obfuscate the sensitive data in your plans. Plan Anonymization will change all Database, Table, and Column names in your plans to generic representations.

There are two ways to use this feature:

- During plan upload you will be presented with the option to anonymize the plan. On the Post to SQLPerformance.com dialog form, select the Anonymize the query plan before upload checkbox option.
- Alternatively, you may anonymize the currently open plan using the Anonymize Button.
found on the toolbar.

- Your original plan will be preserved in the current tab and the anonymized plan will be opened in a new tab.

9.5.1.3 Plan Diagram Options

**Plan Explorer** offers several options and styles which can be applied to the Plan Diagram. When a plan is saved these visual plan properties are maintained as part the Plan Explorer Session. This allows you to create a very specific view of the Plan diagram for both sharing and troubleshooting purposes. For a complete rundown of this feature please see [this blog post](#) from Greg Gonzalez.

**Plan Diagram Toolbar**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoom</td>
<td>Scale the Plan Diagram between 6% - 400% of its original size. You may also use CTRL + mouse scroll wheel to perform this operation.</td>
</tr>
<tr>
<td>Filter</td>
<td>Filter nodes from the Plan Diagram based on Cost.</td>
</tr>
<tr>
<td>Stretch</td>
<td>Stretch the links between Plan Diagram nodes.</td>
</tr>
<tr>
<td>Flatten</td>
<td>Flattens the space between node levels in the Plan Diagram.</td>
</tr>
<tr>
<td>Mode</td>
<td>There are several Modes available which will change the layout of the entire Plan Diagram tree.</td>
</tr>
<tr>
<td>Link Style</td>
<td>The Link Style used between nodes in the Plan Diagram.</td>
</tr>
<tr>
<td>Rotation</td>
<td>Rotates the entire Plan Diagram 90°.</td>
</tr>
<tr>
<td>Defaults</td>
<td>All plan diagram options are reset.</td>
</tr>
</tbody>
</table>

9.5.1.4 Index Analysis

The **Plan Explorer Index Analysis** tab provides a sandbox environment for planning indexing strategy relative to a specific operation in a query. For Index Analysis to be populated, the query must be executed from within the Plan explorer session using the “Get Actual Plan” menu option.

Index analysis involves 5 major areas.

**THE STATEMENT TREE AND PLAN DIAGRAM**

See the [Plan Explorer (integrated)](#) section for more information.

**THE TEXT DATA AREA**

See the [Plan Explorer (integrated)](#) section for more information.
THE INDEXES DATA GRID

The top row is the index score. This is based on several factors including covering, value density, sorting effectiveness, and seekability. 100% is the best possible score, however there are some situations that will prevent the possibility of a 100% score. One known cause is when columns are missing statistics. This will be indicated by the presence of a “?” in several of the column statistics values for a table column.

Column statistics are provided for each table column used by the query, with table columns used by the selected operation displayed in bold.

DATA GRID COLUMNS:

- **Table Column**: Name of the column in the relevant database table
- **Density**: How selective the row values are for the column (lower values are more selective)
- **Last Update**: The date the column statistics were last updated
- **Avg. Length**: Average length of data values
- **Estimated Size (MB)**: The estimated size of the column values (in MB)
- **Output**: Checked if the column is an output for the selected operation
- **Sort Type**: Sort order (ASC, DESC)
- **Sort Order**: Ordinal position of the column in the sort
- **Predicate**: The search or join predicate(s) applied to the table column by the operation

Columns after the “Predicate” column are all index columns. These include both some special case indexes, as well as the indexes defined on the table. They are listed from left to right in the following order:

1. Clustered index (if existing)
2. Index used by the operation (if not the clustered index)
3. Missing indexes as listed in the SQL Server missing indexes DMO
4. Recommended indexes generated by Plan explorer. On occasion, Plan Explorer can recommend a better index
5. The other non-clustered and/or column store indexes that exist for the table, ordered descending by score from left to right

Indexes can be modified, as well as scripted for create, drop, or drop and create. You may also manually update statistics using the button at the bottom of the index column.

INDEX COLORING:

- Shades of green indicate that the ordinal position of the column is such that a favorable operation may occur, such as a seek over a scan. Darker shades of green indicate that the column is filtered.
- Yellow indicates that the column is used in the query, and covered, but is not part of the left subset of the index key. Included columns are also listed in yellow.
- Red indicates that the column involved is not covered by the index.

THE PARAMETERS DATA GRID

The parameters grid lists parameters used by the query. This differs from the Plan explorer Parameters tab in that it uses tipping point logic to help determine the type of operation (SEEK or SCAN) that the optimizer might choose for that parameter value. This is not an exact science, and
should be used as a rough guideline.

The parameters grid provides a “Test Value” column which allows you to enter a new parameter value, and ask for a new estimated plan. This allows you to test different parameter values against the plan to see if the plan shape will change given that parameter value.

The small arrows to the right of parameter values allow you to quickly place the value of that parameter into the Test Values column.

**THE STATISTICS HISTOGRAM CHART**

The histogram chart displays the histogram steps for the index statistics visually. This allows you to quickly spot potential parameter sniffing issues.

This chart also works in conjunction with the parameters grid, and selected parameters that are present in the histogram will have their histogram buckets highlighted on the chart. This also works for parameter ranges.

---

### 9.5.2 Top SQL Runtime Stats

The Top SQL Runtime Stats provides insight into the frequency and duration of the queries and procedures that are captured by Top SQL.

**RUNTIME GRAPHS**

There are two views available for the graph, Runtime History and Runtime Aggregates. This selection can be made from the main toolbar. These graphs make it easy to identify runtime trends, which can be an indicator of or eventually lead to performance issues with the event or the server.

**Runtime History Graph**

This graph shows an entry’s actual runtime and status over the specified interval. A red bar indicates that that instance failed.

**Runtime Aggregates Graph**

This graphs show the distribution of the entry’s actual runtime over the specified interval.

**GRAPH CONTROLS**

Use the Navigation toolbar to change the start date/time and interval for the graph. The Days control can go out up to 60 days at a time. Whenever the selected interval is less than 1 day (1 minute through 4 hours), the Days control will be changed to 1 day.

| Min/Max Y Val | When a runtime graph is first displayed, SentryOne will try to determine the most appropriate interval and maximum Y-axis value. You can change either of these settings at any time. |
### Auto-scale to Max Value
Checking this box will automatically select the appropriate Max Y Value of the data being shown in the current view.

### Interval Type
Changes the interval used for the Y-axis.

### Show Value Labels
Deselect this checkbox to hide the bar value labels. This can become necessary for broader ranges with hundreds or thousands of bars to reduce the noise on the graph.

### RUNTIME GRID

The Runtime Grid provides additional information about captured Top SQL events. This includes both events that are captured as part of the standard Top SQL collection, and fast running events which are captured as part of Query Stats collection. The events listed in the Runtime Grid are not dependent on any specific time range you are viewing, and do not update based on the selected chart view. Instead, the count represents the total number of observed occurrences of the event for which you have history in SentryOne. The respective Min, Max, and Average durations are updated as new occurrences of the events are observed in your environment.

Events are purged from the list according to the Performance Monitor – Purge History Older Than setting. You can see the last time an event occurred by using the Last Execution column. If an event has not been observed since the effective purge history older than date, the record of it will be removed.

### CONFIGURING SETTINGS AND ALERTS FOR INDIVIDUAL EVENTS

In addition to the count and duration metrics, there is also an option to override existing inherited Top SQL Runtime thresholds for specific Top SQL events. To do this, click the Override check box to the right of the value that you would like to override and change the value. These thresholds correspond to the Top SQL: Runtime Threshold Max and Top SQL: Runtime Threshold Min conditions, which can be configured at the Instance level and above. The ability to configure thresholds for specific events, without altering Instance and global level thresholds, allows you to fine tune the alerts you receive.

### NAVIGATION TO THE TOP SQL RUNTIME STATS

To access the Top SQL Runtime Stats, you can right click on a SQL Server Instance and select Jump to Top SQL Runtime Stats. Alternatively, you can right click on an entry in the Top SQL tab and select Jump to Runtime Stats, which will automatically highlight the entry that was selected.

---

For information regarding the Event Manager Runtime Stats, see [Runtime Stats](#).

---

### 9.6 AlwaysOn

**Applies to:** SQL Sentry
The **SQL Sentry AlwaysOn Management** page can be accessed by right-clicking the **All Targets** node in the **Navigator pane** and choosing **Open → AlwaysOn Management**. You may also open **AlwaysOn Management** from the context menu of any **Site** or **Group** level node as applicable. The AlwaysOn tab is also available within SentryOne for any monitored SQL Server instance hosting an availability group replica.

The AlwaysOn Management page is divided into three main areas:

- **The Overview Area** offers several unique actionable views of your AlwaysOn environment, including easily digestible at-a-glance **Status Information**.
- **The History Area** displays both historical charts and AlwaysOn health events from your environment.
- **The Details Gridview Area** displays both high level and detailed level metrics concerning your environment.

You may customize visual aspects of the AlwaysOn views around your unique environment, including the default thresholds for Log and Recovery Queues. See the **Customization section** for more information.

AlwaysOn Management includes full alerting capability surrounding your AlwaysOn environment. Fully customizable **Conditions** are available to alert you on both Health and Failover status. For more information, see the **AlwaysOn Alerting section**.

**SERVER VISIBILITY**

When you monitor the SQL Server instance hosting the **Primary Replica** of an availability group with **SQL Sentry**, the entire topology of that availability group will be displayed as part of the Overview Area. Unwatched instances hosting replicas are displayed with a gray background.

In order to enable complete AlwaysOn monitoring and alerting, each replica in the availability group should be watched with **SQL Sentry**. Status information is not available for unwatched replicas. This includes historical metrics, charts, and data pertaining to any associated Send or Recovery Queues.

To monitor a server right-click the node and choose **Watch** from the context menu.

**OVERVIEW AREA**

The Overview Area contains several layouts for managing your AlwaysOn environment from several perspectives, including: the Windows Server Failover Cluster level, the Availability Group level, the SQL Server Instance level, and the Replica level. The Layout Style can be changed from the dropdown list located in the top corner of the Overview Area.

Status information which is common to several views can be found in the **Status Information** section.

**STATUS INFORMATION**

Each of the Overview Diagrams contains a unique representation of those individual availability replicas/WSFCs which are part of your managed AlwaysOn environment. Regardless of the selected view, these nodes and the pipes connecting them all share a common set of visual status information which is described below.

**Context Menu Options**
Watch – Watch the SQL Server instance hosting the replica with SentryOne Performance Analysis
Add Replica – Opens the native SSMS "Add Replica to Availability Group" dialog
Remove Replica – Opens the native SSMS "Remove Secondary Replica from Availability Group" dialog
Failover – Opens the native SSMS "Fail Over Availability Group" dialog

Node Specific
The nodes convey status and configuration information about the replica, including Queue Health, Connection mode, and overall Health Status.

Labels
Instance Name - On applicable views the node contains a label with the SQL Server instance name.

Availability Replica Role - Each node is labeled indicating the role of the availability replica, as AG Primary, AG Secondary, or FCI Secondary

Indicator Bars
Health Status - The health status or availability group state is represented by the left bar of the node. Green represents a healthy availability group state. Pink represents an unhealthy availability group state. The Health Status bar will also be pink if connectivity has been lost between the SentryOne Monitoring Service and the SQL Server instance.

Queue Indicator - The Log Send Queue / Recovery Queue status is visually represented by the bar on the right. As the respective queue grows the color will change from a dark green, to light green, to orange. Once the Queue KB Threshold has been passed the color will change to red.

Status Boxes
Failover Mode - The failover mode of the availability replica is indicated in the left status box.
  - Automatic Failover mode - represented with a play icon
  - Manual mode - does not contain an icon.

Client Connection Mode - The client connection mode of the availability replica is represented in the center status box. If the availability replica is a readable secondary it will be represented with a small glyph.

Failover Cluster Instances - If the node is a Failover Cluster Instance (FCI) it will be represented in the right status box.

Availability Mode
The availability mode of the availability replica is represented as follows:
  - Asynchronous-commit mode - is represented by two black lines drawn on the pipe between availability replicas.
  - Synchronous-commit mode - is represented by the absence of these two black lines drawn
on the pipe.

Pipe and Color Specific
The visual characteristics of the pipes connecting the nodes convey status information about the health of the relationship between nodes. **Pipe saturation** is an indicator of the amount of information moving between replicas. **Pipe color** is dependent on the recovery queue of the secondary.

Pipe Saturation
The pipe becomes saturated as the amount of data being transferred increases. More explicitly the pipe becomes saturated as the **KB To Replica/sec** value approaches the **KB To Replica/sec threshold**.

By default, the **KB To Replica/sec threshold** value is dynamic and will be the highest global value seen between replicas since first opening the AlwaysOn tab.

The **KB To Replica/sec threshold** can customized with the **AlwaysOn.AvailabilityReplicaTransferThreshold** table in the SentryOne Database. See Also: **Customization**

Pipe Color
The pipe color changes as the recovery queue grows; the color will change from a dark green, to light green, to orange, and finally once the **Recovery Queue KB Threshold** has been passed the color will change to red.

By default the **Recovery Queue KB Threshold** value is 1MB or 1024 KB. The **Recovery Queue KB Threshold** can be customized with the **AlwaysOn.AvailabilityReplica** table in the SentryOne Database. See Also: **Customization**

HISTORY AREA
The History Area contains tabs for Charting, Error Log, and Replica State Changes. Information in each tab is shown based on the selected time range. In live mode, each tab includes the last ten minutes of collected data. Updating the time range on the tool bar will change this displayed range. Each of the History tabs is context aware and will display information based on the node or row selected in either the Overview or Gridview Area. Clicking on the diagram background will display log information for all nodes and replicas.

ALWAYSON HEALTH COLLECTION
The Error Log and Replica State Changes tabs are initially disabled. This is because AlwaysOn health collection is disabled by default to prevent innocuous entries associated with Extended
Events from being repeatedly logged to the SQL Server Error Log without your knowledge or approval. Please see the update below for a link to CU 6 for SQL Server 2012 SP1, which contains a fix for this issue.

If you’d like to enable health collection, click the Enable button on either the Error Log or Replica State Changes tab.

⚠️ **IMPORTANT:** Enabling automated AlwaysOn health (error and state change info) collection will cause innocuous entries associated with Extended Events to be repeatedly logged to the SQL Server Error Log, as described in Microsoft Connect item 783036. We are working closely with Microsoft to correct this unfortunate behavior. Until it is corrected, please note that the entry, “Using ‘dbghelp.dll’ version ‘4.0.5’”, will be sent to the Error Log on all SQL Servers with Availability Groups monitored by SentryOne, at the rate specified by the collection interval.

**UPDATE:** Cumulative Update package 6 for SQL Server 2012 SP1 addresses this issue. Once the update has been applied, innocuous entries associated with Extended Events will no longer be logged to the SQL Server Error Log.

The enabled status and interval can be adjusted at any time from the **SentryOne Monitoring Service** -> **Settings** -> **Event Monitor tab**.

**Charting Area**

Various charts related to your managed AlwaysOn environment are available in the expandable History Area of the AlwaysOn Management tab. The Charting area is context aware and will display charts based on the selected node or row.

For instance, from the **WSFC Node Group/Matrix** layout, if you select a Primary Availability Replica, charts for the **KB to Replicas/sec** and **Log Send Queue KB** metrics will be displayed. Selecting a Secondary Availability Replica will display charts with the **KB to Replicas/sec** and **Recovery Queue KB** metrics.

If you select a WSFC in the **WSFC Members** layout, pertinent metrics for each respective Availability Group hosted on the WSFC will be displayed, including KB To Replicas/sec and KB from Replicas/sec metrics.

**Error Log**

The Error Log allows you to view any **AlwaysOn Health events** that are classified as **errors** in your managed environment. This log is equivalent to viewing AlwaysOn error Health Events accessed from the native AlwaysOn dashboard, but offers a number of advantages.

The Error Log is context aware and will display only errors related to the selected node or row. In live mode, this includes the last ten minutes of collected data. Updating the time range on the tool bar will change this displayed range, allowing you to easily view an historical subset of errors for any particular server.

**Replica State Changes**

The Replica State Changes tab displays **AlwaysOn Health events** that are classified as **Replica State Changes**. This log is equivalent to viewing AlwaysOn availability_replica_state_change health events accessed from the native AlwaysOn dashboard, but offers a number of advantages.

The Replica State Changes tab is context aware and will display only state changes related to the selected node or row. In live mode, this includes the last ten minutes of collected data. Updating the time range on the tool bar will change this displayed range, allowing you to easily view an historical subset of state changes for any particular server.
**AlwaysOn Alerting**

SentryOne AlwaysOn Management includes the ability to configure customizable alerts specific to the status and health of your monitored Availability Groups.

Three Failsafe Conditions are available:

- SQL Availability Group Failover
- SQL Availability Replica Healthy
- SQL Availability Replica Unhealthy

Each of these Conditions exposes several **Condition Filters** which can help you to completely customize your desired alerts. **Configuring Alerts**

Select the node appropriate to the level you would like to configure the action for in the Navigator Pane and then the Failsafe Actions tab in the Conditions pane. If you do not see the Conditions pane, use the View menu → Failsafe Actions.

Next, you will want to click the Add button found in the Conditions pane. This will open the Select Action window. Expand the applicable Object and Condition. Use the check box(s) to select which Actions should be taken in response to this Condition being met. Click the OK button.

For more information about Actions that can be taken when a Condition is met, see the **Actions** topic.

---

**CUSTOMIZATION**

**Customizing Pipe Saturation**

By default, the **KB To Replica/sec threshold** value is dynamic and will be the highest global value seen between replicas since first opening the AlwaysOn tab. The **KB To Replica/sec threshold** can be customized with the **AlwaysOn.AvailabilityReplicaTransferThreshold** table in the SentryOne Database. It acts as a mapping table allowing you to define a specific value/threshold for the maximum allowed bytes (**KB to Replica/sec Threshold**). By default, the table will not have any records, and in this state the threshold will be dynamic as mentioned above.

If you would like to set a new **KB To Replica/sec threshold** value specific for all replicas involved in a WSFC node – WSFC node relationship, you would add a record with the **PrimaryNodeName**, **SecondaryNodeName**, and **MaxBytesSentToSecondaryPerSec** values defined. The WSFC node names can be found in the AlwaysOn.ClusterNode and AlwaysOn.AvailabilityReplica tables.

<table>
<thead>
<tr>
<th>ID</th>
<th>PrimaryNodeName</th>
<th>SecondaryNodeName</th>
<th>MaxBytesSentToSecondaryPerSec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you would like to set a new Global **KB To Replica/sec threshold** value, you can add a record with just a **MaxBytesSentToSecondaryPerSec** value. For instance, if you wanted to set a Global value of 5 KB:

---Set a Global Value of 5KB

**INSERT INTO** [SentryOne].[AlwaysOn].[AvailabilityReplicaTransferThreshold]

(MaxBytesSentToSecondaryPerSec)
The new Global **KB To Replica/sec threshold** value is now 5 KB.

If you would like to set a new **KB To Replica/sec threshold** value specific to a Primary Replica - Secondary Replica relationship, you would add a record with the `PrimaryNodeName`, `SecondaryNodeName`, and `MaxBytesSentToSecondaryPerSec` values defined.

---

**Set a Value specific to a Relationship**

```sql
INSERT INTO [SentryOne].[AlwaysOn].[AvailabilityReplicaTransferThreshold]
VALUES ('MyPrimary', 'MySecondary', 5120);
```

The **KB To Replica/sec threshold** value for the specific 'MyPrimary' – 'MySecondary' is now 5KB. The default global value will be used for all other replicas.

If you would like to set a new **KB To Replica/sec threshold** value for a specific Primary Replica, to be used with all of its Secondary Replicas, you would add a record with just the `PrimaryNodeName` and `MaxBytesSentToSecondaryPerSec` values defined.

---

**Set a Value Specific to a Primary Replica to be used with all of its Secondaries**

```sql
INSERT INTO [SentryOne].[AlwaysOn].[AvailabilityReplicaTransferThreshold]
VALUES ('MyPrimary', Null, 5120);
```

The **KB To Replica/sec threshold** value for all Secondary replicas in a relationship with the 'MyPrimary' replica is now 5 KB.

You may add multiple records to the **AlwaysOn.AvailabilityReplicaTransferThreshold** table, allowing you to easily define **KB To Replica/sec thresholds specific to your environment**.

### Customizing Pipe and Queue Indicator colors

By default, there is a Global **Recovery Queue and Log Send Queue KB Threshold** value of 1MB or 1024 KB. These values can be customized with the **AlwaysOn.AvailabilityReplica** table in the SentryOne Database using the `LogSendQueueKBMax` and `RecoveryQueueKBMax` columns.
Each Replica in your monitored environment has an entry in the `AlwaysOn.AvailabilityReplica` table. By default, the `LogSendQueueKBMax` and `RecoveryQueueKBMax` columns will be null. When this is the case, the default Global value (1MB) will be used for each respective Queue. You can set these columns to your desired value for each of your replicas.

For example, if I wanted to update the Log Send Queue KB Threshold value for my replica named 'MyPrimary' to be 10 MBs, I could update the table as follows:

```
--Set a new Log Send Queue KB Threshold value for my replica 'MyPrimary'
Update [SentryOne].[AlwaysOn].[AvailabilityReplica]
set
LogSendQueueKBMax = 10240
Where ReplicaServerName = 'MyPrimary'
```

The replica 'MyPrimary' will now have a Log Send Queue KB Threshold value of 10240 KB or 10 MB.

<table>
<thead>
<tr>
<th>ID</th>
<th>ReplicaServerName</th>
<th>LogSendQueueKBMax</th>
<th>RecoveryQueueKBMax</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MyPrimary</td>
<td>10240</td>
<td>Null</td>
</tr>
</tbody>
</table>

9.7 Query Plans

**Applies to:** SQL Sentry and DB Sentry

The Query Plans tab lists all plans collected for Top SQL events for the specified date range. This provides a detailed chronology of all query plan changes, so you can go back to any point in time to see when a plan change may have led to a query performance problem.

- The "Show Totals" toolbar button controls whether or not the grid lists plans grouped by stored procedure.
- You can jump to this tab by highlighting a range on the dashboard as usual. For example, if you see a spike in parallelism waits on the SQL Server Waits chart, highlight the spike and select Jump To -> Query Plans, then sort by the "Parallel Operations" column to determine which queries likely caused the spike. The same approach can be used to find queries with key or RID lookups since "Lookups" are now shown on the dashboard, as well as other metrics that correlate such as I/O and table/index scans.

**DISPLAY PANES**

The filter pane allows you to filter the Query Plans by SQL Server.

The bottom pane lists the records matching the current filters. There are two modes for this pane: Default and Totals.

The Default mode lists presents all events for the active date range in a standard list format,
sorted in order by DatabaseName, ObjectName, and StatementCreationTime (descending).
The Totals mode lists presents all events for the active date range in a standard list format, sorted in order by DatabaseName, ObjectName, and PlanCreationTime (descending).

**CONTROLS**

**TOP PANE**
- Filter (minimized by default) - After changing the filter, it is necessary to click the Refresh button, or press F5, to apply the filter.

**BOTTOM PANE**
- To open the plan in the Query Plan Analysis window, select the View button under the Plan column (depending on whether the "Show Totals" button is selected, you will need to expand the Database Name to see the Plan column).

**NOTE:** The context menu provides an option to "Highlight recompiles in the current date range" for a statement by enabling the checkbox.

## Indexes tab

Applies to: SQL Sentry

### INDEXES TAB

The Indexes tab displays information that has been collected about your tables and indexes by SentryOne Fragmentation Manager. The Indexes tab can be used to help you make intelligent decisions about index management in your environment: such as when and how to perform defragmentation operations, when to adjust fill factors, or even when an index definition should be changed.

#### Indexes Tab Sections

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview Charts</td>
<td>The Overview chart section, located at the top of the tab, displays a set of charts which contain aggregate statistics about all the indexes analyzed by Fragmentation Manager.</td>
</tr>
<tr>
<td>Index Grid View</td>
<td>The Index Grid View section, located in the middle portion of the tab, allows you to view index related statistics in a tabular format.</td>
</tr>
<tr>
<td>Details Charts</td>
<td>The Detail chart section, located at the bottom of the tab, displays a set of charts which display statistics unique to the individual index. Selecting an index in the grid view area will update the detail level charts.</td>
</tr>
<tr>
<td>Filter</td>
<td>The Filter section, located at the top of Indexes tab, is collapsed by default. It can be used to filter what is shown in the grid view section. You can filter by SQL Server or Database, and you can also control the number of rows shown in the grid view with the Show Top option.</td>
</tr>
</tbody>
</table>

You can easily customize the layout of the Indexes tab. To do so, right click any section heading and choose the Customize Layout option.

#### OVERVIEW CHARTS

The Overview Charts contain aggregate statistics about all of the indexes analyzed on the Target by Fragmentation Manager. For information about each specific chart and related metrics see the sections below.

#### TOTAL FRAGMENTATION

The Total Fragmentation chart displays a once a day aggregation of the fragmentation levels for all indexes that have been analyzed by Fragmentation Manager. Each index’s fragmentation level for the day is summarized, and the indexes are next grouped by range: <10%, 10-20%, 20-30%, and >30%.

The Y-axis numbers represent the total number of indexes belonging to the Target that have been analyzed by Fragmentation Manager. When you select a range on the chart, the indexes which fall into the corresponding range will also be highlighted in the index grid view.

#### TOTAL SPACE USAGE (MB)

The Total Space Usage chart summarizes disk usage for each index which has had data collected over the specified time frame. As a
reminder, you can adjust the minimum and maximum index size collection thresholds in the Database Source Settings.

The chart has individual representations for Unused space, Used space, and Empty space. Unused space is represented at the bottom of the chart, Used space in the middle, and Empty space at the top. See the table below for the explanations of each metric.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unused</td>
<td>Unused space is the amount of disk space which has been allocated for index pages, but does not contain any index pages.</td>
</tr>
<tr>
<td>Used</td>
<td>Used space is the amount of disk space which has been allocated for index pages, and does contain index pages.</td>
</tr>
<tr>
<td>Empty</td>
<td>This is the amount of empty or unfilled space, within any index pages. Lack of page fullness could indicate that index fill factors need to be adjusted.</td>
</tr>
</tbody>
</table>

**TOTAL BUFFER USAGE (MB)**

The Total Buffer Usage chart summarizes SQL Server buffer usage, during the specified time range, for each index that has had data collected. There are individual representations of Used space and Empty space.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>Represents the size of the Used Index pages contained within the SQL Server Buffer pool.</td>
</tr>
<tr>
<td></td>
<td>Represents the amount of empty or unfilled space, within those index pages that reside in the SQL Server Buffer Pool.</td>
</tr>
<tr>
<td>Empty</td>
<td>The Empty Buffer metric can be directly mapped to the Empty disk metric. This is because a page that is only 50% full on disk will also be only 50% full when it is contained in the Buffer.</td>
</tr>
</tbody>
</table>

**DETAIL CHARTS**

The Detail Charts contain statistics about individual indexes analyzed on the Target.

**INDEX FRAGMENTATION**

The Index Fragmentation chart shows the fragmentation percentage of the selected index, measured during each scheduled Fragmentation Scan. If a reorganization or rebuild operation was performed on the index, and you have the Run Post-defrag Analysis setting enabled, you will also see data points which represent the fragmentation percent after the rebuild or reorganization operation is complete. This chart can be used to help you determine what kind of fragmentation management schedule should be applied to the index.

As mentioned, each datapoint on the chart represents a Fragmentation Manager operation. You may double-click on any datapoint to open the Event Calendar for the selected operation. The background colors of the chart correspond to the same ranges and colors as found in the Total Fragmentation chart.

**INDEX SPACE USAGE**

The Index Space Usage chart shows how much of the selected index was located on disk and in buffer, over the selected time frame. It can also give you an idea about how much disk and buffer space is being wasted by the index, due to non-full pages.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unused Disk</td>
<td>The amount of disk space which has been allocated for index pages, but does not contain index pages.</td>
</tr>
<tr>
<td>Used Disk</td>
<td>Used space is the amount of disk space which has been allocated for index pages, and does actually contain index pages.</td>
</tr>
<tr>
<td>Empty Disk</td>
<td>This is the amount of empty or unfilled space, within any of the Index pages.</td>
</tr>
<tr>
<td>Empty Buffer</td>
<td>Represents the amount of disk space taken up by empty or unfilled space, within those index pages, that reside in the SQL Server Buffer Pool.</td>
</tr>
<tr>
<td>Used Buffer</td>
<td>Used Buffer represents the size of the Used Index pages contained within the SQL Server Buffer pool for the specified index.</td>
</tr>
</tbody>
</table>

**INDEX ACTIVITY**

The Index Activity chart gives you a view of how the selected index is being used over the specified time frame. You can use this chart to help identify both unused indexes and the cost of your heavily used indexes.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Scans</td>
<td>The number of scan operations on the index caused by user activity.</td>
</tr>
<tr>
<td>User Seeks</td>
<td>The number of seek operations on the index caused by user activity.</td>
</tr>
</tbody>
</table>
The number of lookups caused by user activity.

- For a clustered index, this number represents the number of key lookups done against the index.
- N/A for non-clustered index
- For a heap, this number indicates the number of RID lookups.

The number of updates caused by user activity. Indicates the level of maintenance on the index caused by insert, update, or delete operations done to the underlying table.

The number of scan operations on the index due to system queries.

For general information about the Index usage statistics see the `sys.dm_db_index_usage_stats` MSN topic.

INDEX GRID VIEW

The Index Grid View, located in the center of the Indexes tab, contains various statistics about your individual indexes. When you select an index in the grid view the Detail index charts will be updated with information specific to that index.

You can apply filters and order the data by various columns. To apply a filter to the grid view, mouse over a column and select the filter symbol located in the upper right hand corner of the column header. After you apply a filter, the filter symbol will persist in the column's header, allowing you to easily see which columns have filters applied. Additionally, the column name and any associated filtered text will be displayed at the bottom of the grid view. Use the X button located beside the filtered text to remove the filter, or use the drop down box to cycle through filtered columns. You can also edit the applied filters with the Edit Filter button found in the bottom of the grid view. Filters can also be applied through the right click context menu of any column.

The information displayed in the Overview Charts is not effected by filters applied in the Grid view.

To order the grid view by a column's value, simply select the column header. An arrow will be shown in the column's header indicating that the grid view is being ordered by that column. Cycle through ascending and descending order by selecting the column's header.

**Index Grid View Columns**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server</td>
<td>The name of the SQL Server where the index resides.</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>The name of the Database where the index resides</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>The name of Table where the index resides</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>The name of the Index.</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>The Type of Index. Heap, Clustered Index, Non-clustered Index, Primary Index, Spatial Index, or XML Index. For general information about the different types of indexes see the Types of Indexes MSN article.</td>
<td></td>
</tr>
<tr>
<td>Primary Key</td>
<td>Specifies if the index is part of a primary key constraint.</td>
<td></td>
</tr>
<tr>
<td>Unique</td>
<td>Specifies if the index is part of a unique constraint.</td>
<td></td>
</tr>
<tr>
<td>Data Space Type</td>
<td>Specifies the Data Space Type.</td>
<td></td>
</tr>
<tr>
<td>Size (MB)</td>
<td>The amount of disk space represented in megabytes taken up by the index.</td>
<td></td>
</tr>
<tr>
<td>Used (MB)</td>
<td>Used space is the amount of disk space which has been allocated for index pages, and does contain index pages.</td>
<td></td>
</tr>
<tr>
<td>Used (MB) in Buffer</td>
<td>Represents the size of the Used Index pages contained within the SQL Server Buffer pool.</td>
<td></td>
</tr>
<tr>
<td>Empty (MB) in Buffer</td>
<td>Represents the amount of empty or unfilled space, within those index pages that reside in the SQL Server Buffer Pool.</td>
<td>Not available for Limited Scan mode.</td>
</tr>
<tr>
<td>Row Count</td>
<td>The number of rows of data belonging to the index.</td>
<td>Not available with a Sampled Scan for heaps.</td>
</tr>
<tr>
<td>Avg % Fragmented</td>
<td>The average fragmentation percentage of the index as recorded during the last fragmentation scan.</td>
<td>Not available for Limited Scan mode.</td>
</tr>
<tr>
<td>Avg % Page Space Used</td>
<td>The average percentage of available data storage space used in all the pages of the selected index. This indicates page fullness.</td>
<td></td>
</tr>
<tr>
<td>Fill Factor</td>
<td>The Fill Factor set for the selected index. For more information see the Specify Fill Factor for an Index MSN topic.</td>
<td></td>
</tr>
</tbody>
</table>
### User Scans Delta

The number of scan operations on the index caused by user activity since the last sample time (every 15 minutes).

### Status

The current status of any defragmentation operations being performed on the selected index.

### Start Time

The start time of the last defragmentation operation that was performed on the selected index.

### End Time

The end time of the last defragmentation operation that was performed on the selected index.

### Duration

The duration of the last defragmentation operation that was performed on the selected index.

### Last Message

Indicates the last defragmentation operation related to this index, including the last defragmentation method used on the selected index.

### Avg Record Size (bytes)

Indicates the average record size in bytes for the selected index.

### Compressed Pages

The number of compressed pages for the selected index.

### Enabled

Indicates if the Index has been made available. Set in the index properties of the native SQL Server tools.

### Forwarded Records

Represents the number of records in a heap that have forward pointers to another data location. Forward pointers can be added during updates, when there is not enough room to store the new row in the original location.

Number of ghost records ready for removal by the ghost cleanup task in the allocation unit. Ghost records are records that have been logically deleted from a page but not physically deleted.

For more information about Ghost Records see these two SQL Skills articles by Paul Randal:

- Inside the Storage Engine: Ghost cleanup in depth
- Ghost cleanup redux

### Ghost Records

Not available with a Limited Scan.

### Pages

The total number of index pages for the selected index.

### Unused Disk (MB)

The amount of disk space which has been allocated for index pages, but does not contain index pages.

### Used %

The percent of disk space which has been allocated for index pages and contains index pages.

### Version Ghost Records

Number of ghost records retained by an outstanding snapshot isolation transaction in an allocation unit.

Not available with a Limited Scan.

### LOB Data

Specifies if the index contains one or more LOB data type.

### LOB Data

Not available with a Limited Scan.

### INDEX GRID VIEW CONTEXT MENU COMMANDS

As applicable the following commands are available through the right-click context menu of the Index Grid and Tree View.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Properties</td>
<td>Use the Index Properties command to view the Index Properties of the selected index in the native SQL Server tools.</td>
</tr>
<tr>
<td>Analyze Fragmentation Now</td>
<td>Selected your desired mode to perform an Fragmentation analysis on the selected item.</td>
</tr>
<tr>
<td>Sample mode</td>
<td>returns statistics based on a 1 percent sample of all the pages in the index or heap. If the index or heap has fewer than 10,000 pages, Detailed mode is used instead of Sampled.</td>
</tr>
<tr>
<td>Detail mode</td>
<td>scans all pages and returns all statistics.</td>
</tr>
<tr>
<td>Use the Rebuild (offline), Rebuild (online), or Reorganize command to immediately perform the selected operation.</td>
<td>For general information about Scan Modes see this MSDN article.</td>
</tr>
<tr>
<td>Defrag or Analysis</td>
<td>For general information about reorganizing and rebuilding indexes see this MSDN article.</td>
</tr>
<tr>
<td>Exclude Index from Automated Defrag or Analysis</td>
<td>The Exclude Index from Automated Defrag or Analysis command is used on indexes which contain LOB columns offline rebuild will be performed.</td>
</tr>
<tr>
<td>Reset Fragmentation Stats</td>
<td>The Reset Fragmentation Stats command will reset statistics for the selected item, including the Avg % Fragmented statistic.</td>
</tr>
<tr>
<td>Jump To Navigator</td>
<td>The Jump To Navigator command will open the selected item in the Navigator pane.</td>
</tr>
</tbody>
</table>
9.8.1 Fragmentation Manager

Heavily fragmented indexes can degrade the performance of your database and the applications running on it. Index fragmentation can be resolved by reorganizing or rebuilding an index. Fragmentation Manager can automatically collect table and index information, analyze the data, take the appropriate reorganization or rebuild actions, and then perform post defragmentation analysis.

Fragmentation Manager has a dedicated tab in SQL Sentry, Indexes. The Indexes tab displays index related statistics and charts, from the Target level down to the individual index level, giving you a complete view of the fragmentation levels on your server. Having this information allows you to make intelligent decisions about index management in your environment; such as when and how to perform defragmentation operations, when to adjust fill factors, or even when an index definition should be changed.

Different schedules can be set for different Instances and databases, even down to the individual table or index level, allowing you complete granular control over any defragmentation actions. Additionally, you have the ability to set specific schedules for rebuilds or reorganizations explicitly. Several additional settings are available to help you fine tune the actions SentryOne takes, including:

- The ability to set the scan level or mode that is used to obtain fragmentation statistics.
- The ability to set both minimum and maximum index size thresholds for the collection of fragmentation data.
- The ability to set reorganization and rebuild fragmentation threshold percentages.
- All index defragmentation settings work within the normal SentryOne hierarchy, so settings can be configured at one level and be automatically inherited by objects below it, allowing for easy automation within your environment.

For a complete explanation of all the available settings, see the Fragmentation Manager Settings section below.

**ENABLING FRAGMENTATION MANAGER**

You can enable Fragmentation Manager through the right-click context menu of any Instance or by opening the Indexes tab of SQL Sentry and selecting the Enable Now button. The first time you enable Fragmentation Manager a Wizard will be displayed.

**Note**: All Fragmentation Settings will become available through the Settings pane after running the Wizard.

### Fragmentation Manager Wizard Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect table and index size information</td>
<td>The base level of Fragmentation Management. Every 15 minutes SentryOne will collect table and index size information for the top 500 indexes in each database.</td>
</tr>
<tr>
<td>Collect buffer data when buffer size &gt; 8GB</td>
<td>Determines if SentryOne will collect Buffer data when buffer size is &gt; 8 GB. See the Database Source Setting Collect Buffer Data when Buffer &gt; 8GB description for a complete explanation.</td>
</tr>
<tr>
<td>Retain historical data [FM]</td>
<td>Table and index historical data will be retained.</td>
</tr>
<tr>
<td>Analyze fragmentation</td>
<td>SentryOne will analyze index fragmentation statistics based on the schedule you specify on the next screen and display those statistics on the Indexes tab of SQL Sentry. The Limited Mode will be used for obtaining fragmentation statistics. The Fragmentation Scan Mode may be changed after the wizard completes from the Index Defragmentation Settings.</td>
</tr>
<tr>
<td>Defragment indexes over specified thresholds [FM]</td>
<td>SentryOne will perform defragmentation operations based on the schedule you specify in the next screen. Select your desired reorganization and rebuild thresholds for defragmentation operations.</td>
</tr>
<tr>
<td>Reanalyze indexes after defragmentation</td>
<td>SentryOne will reanalyze fragmentation statistics after any defragmentation operations are performed.</td>
</tr>
</tbody>
</table>

### Select Schedule

To select a schedule to be used for analysis and/or defragmentation, you may either choose a pre-existing schedule, or use the
New command to create a new schedule. Select Next to confirm your settings and then Finish to complete the Wizard.

FRAGMENTATION MANAGER RELATED SETTINGS

There are two groups of settings relevant to Fragmentation Manager. Database Source Settings are used to configure the general collection of table and index information, including size collection thresholds, buffer collection thresholds, and index partition options. Index Defragmentation Settings are used to configure the defragmentation and analysis operations, including scheduling, and setting index reorganization and rebuild thresholds.

DATABASE SOURCE SETTINGS

<table>
<thead>
<tr>
<th>Synchronization</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit From Parent</td>
<td>Specifies whether settings in this group are being inherited. For more information about inheritance, see the hierarchy chart in the Alerting and Response System topic.</td>
<td>True</td>
</tr>
<tr>
<td>Maximum Rows to Synchronize</td>
<td>Maximum rows of historical event data to collect from this source.</td>
<td>5000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General Settings</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit From Parent</td>
<td>Specifies whether settings in this group are being inherited. For more information about inheritance, see the hierarchy chart in the Alerting and Response System topic.</td>
<td>True</td>
</tr>
<tr>
<td>Enable Table/Index Data Collection</td>
<td>Specifies whether table and index collection is enabled. When enabled every 15 minutes SentryOne will collect table and index size information.</td>
<td>True</td>
</tr>
</tbody>
</table>

**Note:** This Setting must be True in order to perform Analysis operations and Automated Defragmentation operations.

Max Partitions to collect per Database

The maximum number of partitions for which information will be collected per database.

Min Index Size (MB) to Collect Fragmentation Data

Specifies how large an index must be before statistics are captured about it. If you keep the default value of 10 MB any indexes which are smaller than 10 MB will not be analyzed or considered for defragmentation operations.

Max Index Size (MB) to Collect Fragmentation Data

Sets the maximum size an index can be, and still be considered for defragmentation operations.

Collect Buffer Data when Buffer > 8GB

Collecting detailed buffer information is inherently a low overhead process. However, since it takes approximately 1 second per GB of buffer, on larger buffers the associated query can take some time to complete, which may cause it to appear in Top SQL and/or generate associated alerts.

It is generally safe to enable this setting, since SentryOne ensures that the total time spent per day collecting buffer data will be roughly the same regardless of buffer size, and collection only occurs when buffer has changed.

When collecting buffer data, SentryOne uses a variable collection frequency, which is based on the size of the buffer. The chart below shows the variable collection frequency along with the approximate time it takes to collect buffer data for different buffer sizes.

**Maximum Buffer Data Collection Frequency**

<table>
<thead>
<tr>
<th>Buffer Size (GB)</th>
<th>Collection Frequency (Min)</th>
<th>Time to Collect (Sec)</th>
<th>Time to Collect (Min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>15</td>
<td>4</td>
<td>0.1</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>8</td>
<td>1.1</td>
</tr>
<tr>
<td>16</td>
<td>30</td>
<td>16</td>
<td>0.3</td>
</tr>
<tr>
<td>32</td>
<td>60</td>
<td>32</td>
<td>0.5</td>
</tr>
<tr>
<td>64</td>
<td>120</td>
<td>64</td>
<td>1.1</td>
</tr>
<tr>
<td>96</td>
<td>240</td>
<td>96</td>
<td>1.6</td>
</tr>
<tr>
<td>128</td>
<td>240</td>
<td>128</td>
<td>2.1</td>
</tr>
<tr>
<td>256</td>
<td>480</td>
<td>256</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Set with Wizard
INDEX DEFRAGMENTATION SETTINGS

After you enable Fragmentation Manager the Index Defragmentation Settings can be accessed through the Settings pane. Index Defragmentation Settings can be configured at the following levels: Global (All Targets), Sites, Target Group, Target, Instance, Database, Table, or at the individual index.

For instance, if you wanted to configure the Index Defragmentation Settings for a specific Instance, first select the Instance in the Navigator, and then open the Settings pane. If you do not see the Settings pane, use the View Menu (View -> Settings). From here use the drop-down menu and choose Index Defragmentation.

Below you will find a table with a description of the various Index Defragmentation Settings and the default values.

<table>
<thead>
<tr>
<th>General Defrag Settings</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit From Parent</td>
<td>Specifies whether settings in this group are being inherited. For more information about inheritance, see the hierarchy chart in the Alerting and Response System topic.</td>
<td>False</td>
</tr>
<tr>
<td>Primary Schedule</td>
<td>The schedule on which index analysis and/or defrag will be performed, according to the configured reorg and rebuild thresholds. For more information about Schedules, see the Schedules topic.</td>
<td>&lt;not specified&gt;</td>
</tr>
<tr>
<td>Operation Type</td>
<td>Specifies whether analysis only or analysis and defrag operations occur under the Primary schedule.</td>
<td>Analyze Only</td>
</tr>
<tr>
<td>Reanalyze Indexes after Defrag</td>
<td>Specifies whether to run the fragmentation analysis again after any scheduled defrag.</td>
<td>False</td>
</tr>
<tr>
<td>Fragmentation Scan Mode</td>
<td>The Scan mode used for obtaining fragmentation statistics. <strong>Limited mode</strong> is the fastest mode and scans the smallest number of pages. For an index, only the parent-level pages of the B-tree are Scanned. For a heap, only the associated PFS and IAM pages are examined; the data pages of the heap are not scanned. <strong>Sample mode</strong> returns statistics based on a 1 percent sample of all the pages in the index or heap. If the index or heap has fewer than 10,000 pages, <strong>Detailed mode</strong> is used instead of <strong>Sampled</strong>. <strong>Detail mode</strong> scans all pages and returns all statistics. For general information about Scan Modes see this MSDN article.</td>
<td>Limited</td>
</tr>
<tr>
<td>Maximum Concurrent Operations</td>
<td>Specifies the maximum number of concurrent operations. Operations are defined as an analysis, rebuild, or reorganization. This setting is capped at 5.</td>
<td>1</td>
</tr>
<tr>
<td>Delay Between Defrag Operations</td>
<td>When Maximum Concurrent Operations is set to 1, you can specify a delay between defrag operations to prevent send/redo queue overload when availability groups or mirroring are in use.</td>
<td>30 seconds</td>
</tr>
<tr>
<td>Maximum Duration</td>
<td>This sets the maximum allowed runtime for the defragmentation process. If the maximum duration is met during a rebuild operation the rebuild will still be completed. If the maximum duration is met during a reorganization operation the reorganization will be stopped and started again during the next scheduled cycle.</td>
<td>2 hours</td>
</tr>
<tr>
<td>Partitions</td>
<td>Partitions to include in the rebuild. This option is applicable for indexes that have a sliding windows partition scheme. For more information about partition schemes see this MSDN article.</td>
<td>All Partitions</td>
</tr>
<tr>
<td>All</td>
<td>All partitions belonging to the index which meet the defragmentation criteria will be either rebuilt or reorganized.</td>
<td></td>
</tr>
<tr>
<td>Max Only</td>
<td>If the maximum numbered partition of the index meets the defragmentation criteria it will be rebuilt or reorganized.</td>
<td></td>
</tr>
<tr>
<td>Exclude Max</td>
<td>All partitions which meet the defragmentation criteria, excluding the maximum numbered partition, will be either rebuilt or reorganized.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Online rebuilds for partitioned indexes can only be performed if the Partitions setting is All and all partitions in the index meet the defragmentation criteria. If the above criteria is not met an Offline Rebuild will be performed.
Regarding the space usage of indexes. For more information, please refer to the Disk Space topic.

By enabling Fragmentation Manager, the functionality of the Disk Space tab is also enhanced by providing additional information regarding the space usage of indexes. For more information, please refer to the Disk Space topic.

## Manual Fragmentation Operations

Fragmentation operations can also be initiated manually. Within the Navigator you can use the right-click context menu of any database, table, or index to initiate fragmentation operations, including analysis, reorganizations, or rebuilds.

Manual fragmentation operations can also be initiated within the Indexes tab. From the Grid/Tree view found in the center of the screen, use the context menu of any database, table, or index to access fragmentation operations.

### Fragmentation Alert Conditions

The following fragmentation related Conditions are available to configure Actions for:

- Defragmentation Completed
- Defragmentation Started
- Defragmentation Failure

To configure a Fragmentation related Condition first select the node appropriate to the level you would like to configure the Action for in the Navigator Pane and then open the General Actions section in the Conditions pane. If you do not see the Conditions pane use the View menu ➔ Conditions.

Next you will want to click the Add button found in the Conditions pane. This will open the Actions Selector window. Expand the Index Actions and then the appropriate Condition. Use the check box(s) to select which Actions should be taken in response to this Condition being met. Click the OK button.

For more information about Actions that can be taken when a Condition is met see the Actions topic.

### Indexes Tab

For information about the charts and statistics displayed in the Indexes tab see the Indexes topic.

### Disk Space Usage Band

By enabling Fragmentation Manager, the functionality of the Disk Space tab is also enhanced by providing additional information regarding the space usage of indexes. For more information, please refer to the Disk Space topic.
SSAS Usage Totals

**Applies to:** BI Sentry

The **SSAS Usage Totals** tab provides information that can be used to determine what actions to take based on activity levels. The information is divided into three different groups:

**Attributes:**

This tab is used to see which attribute combinations are hit the most to determine if creating an aggregation would improve performance.

**Aggregations:**

This tab shows the aggregations that are being used.

**Partitions:**

The Partition tab shows which partitions are being read the most.

**Display Panes**

The **Filter** pane provides a series of filters and controls for specifying which records to display. The available filters are:

- SQL Servers
- Databases
- Cubes
- Measure Groups

**CONTROLS**

- Filter (minimized by default) - After changing any of the filters, it is necessary to click the Refresh button, or press F5, to apply the filter.
- Click on a combination of items in any of the lists to set the filters. The default filter is to display all records.
- To select more than one item in a list, use shift-click or ctrl-click.
- Show Top - Depending on the length of the displayed interval, you may have a lot of data to work with. This setting controls how many of the rows are actually retrieved.

Blocking SQL

**Applies to:** SQL Sentry

The **Blocking SQL** tab displays all SQL Server blocks that occurred during the active date range, which meet the Minimum Block Duration. Each block is displayed in a hierarchical format, showing the relationships between all blocking and blocked SPIDs in a blocking chain.

**BLOCKING SQL COLLECTION**

Blocking SQL collection is controlled with the **Blocking SQL Source Settings**. The **Minimum Block Duration** setting determines how long a block must exist before information is collected on it. By default this is 15 seconds. You may adjust Blocking Collection as desired through the Settings pane (**View menu → Settings**).
Expand for Blocking Settings Examples

If you wanted to change blocking collection to 20 seconds **Globally**:

1. Select the **Shared Groups** node in the **Navigator Pane** (Global context).
2. In the **Settings pane**, use the top drop-down list and select **SQL Server Settings**.
3. Use the second drop-down list to select **Blocking SQL Source**. You should now see the Blocking Source Settings that are being applied Globally.
4. Change the **Minimum Block Duration Setting** to **20 seconds**, it will be saved automatically.

If you wanted to change blocking collection for an individual **Instance**:

1. Select the desired **Instance** node in the **Navigator Pane**.
2. In the **Settings pane**, use the drop-down list and select **Blocking SQL Source**.
3. You should now see the Blocking SQL Source Settings which are configured for the Instance. Change the **Inherit From Parent Setting** to **False**.
4. Change the **Minimum Block Duration Setting** to your desired value, it will be saved automatically.

**BLOCKING SQL RETENTION**

By default, Performance Analysis Blocking SQL, Top SQL, and Deadlock data is retained for 15 days. This is controlled with the **Keep Performance History For Setting** found on the Storage tab of the Global Settings (**Navigator pane → Configuration → Global Settings → Storage tab**). For more information about retention see the [Data Capacity Planning KB article](#).

**BLOCKING SQL ALERTING**

You may configure Actions in response to a number of Blocking SQL Conditions. The **SQL Server: Blocking SQL Condition** occurs each time a block is collected, and therefore has a direct correlation to the Blocking SQL Source collection Settings. You may also configure an **Action** in response to the **SQL Server: Blocking SQL: Output Content Match (OCM) Condition**. The **OCM Condition** can be useful in narrowing the notifications you see concerning blocking.

You may want to consider applying a **Response Ruleset** to any Action you configure with the **SQL Server: Blocking SQL Condition**. **Response Rulesets** allow you to control how often Actions are taken in response to Conditions. In this case, you could apply a Response Ruleset to limit the Blocking alerts you receive, while still retaining Blocking information on the Blocking SQL tab.

**SQL TEXTDATA LENGTH**

By default textdata for captured events is truncated at 10,000 characters. You can change this maximum length in the **Monitoring Service Settings** (**Navigator pane → Configuration → Global Settings → Advanced tab**).

**DISPLAY/CONTROLS**

**TOOLBAR**

The Blocking SQL tab has two modes, **Real Time** and **History**. The active mode is controlled through the toolbar. When the auto-refresh **Play** button is selected, the tab enters **Real Time** mode. In Real Time mode, any blocking SQL which happened within the last 10 minutes will be
When the auto-refresh Pause button is selected, the tab goes into History mode. Select a time range and then use the Go toolbar button to view blocks which happened during that time period. Alternatively, use the Jump To Last Block button to navigate to the last block.

**FILTER**

The top pane provides client filters for specifying which records to display for the defined interval.

```
Note: The Filter pane is for Client side filtering and does not impact what is actually collected. See the Blocking SQL Collection section of this topic for collection related information.
```

After changing any of the filters, it is necessary to click the Refresh button, or press F5, to apply the filter. To select more than one item in a list, use shift-click or ctrl-click. The available filters are:

- SQL Servers
- Applications
- Databases
- Hosts
- Wait Resources

Show Top - Depending on the length of the displayed interval, you may have a lot of data to work with. This setting controls how many of the rows are actually retrieved.

**BLOCKING SQL GRIDVIEW**

Each block is displayed in a hierarchical format, showing the relationships between all blocking and blocked SPIDs in a blocking chain.

The head of the blocking chain will be the top node in the hierarchy. This will contain the blocking statement. It will be displayed with an orange node if the statement is still running, and a green node once it has completed. Sub-nodes in the chain are the statements which are being blocked, and they are displayed with a red node. Any subsequent statements which are being blocked will be nested beneath, giving you a complete picture of the blocking chain.

A unique record is shown for each version of a blocking chain, denoted by the Version column. A new record is created every time the blocking chain changes, which means that a blocked SPID was either added or removed from the chain between polling intervals. For some blocks this may happen frequently causing multiple records to be created, while others may not change at all for the duration of the block. The polling interval for blocks is determined by the Event Monitor Polling Interval (Navigator pane → Monitoring Service node → Settings → Event Monitor tab).

**TEXT DATA**

The bottom pane shows the TSQL text data for the currently highlighted Blocking SQL record. Note that for some SPIDs this may be blank if SentryOne was unable to collect the TSQL due to timing issues.

**ADDITIONAL OPTIONS**

There are additional options available through the right-click context menu, including:

- Jump to Calendar - go to that event on the Event Calendar (requires server to be watched with EM).
- Jump to Dashboard - open the Performance Analysis Dashboard with the selected **Blocking event overlaid** on each of the charts.
- Jump to Top SQL - open the Top SQL tab highlighting the query involved in the blocking event.
- Right-clicking on a blocking record presents a context menu that will allow you to jump to the instance on the calendar or kill the process associated with the blocking event.
- Use the Clear Sort command available through the context menu to remove any grid orderings.
- Additional context menu options include the ability to expand and collapse individual trees or all of the records in the gridview.

## BLOCKING SQL METRICS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPID</strong> [ecid]</td>
<td>The session process ID of the associated blocked/blocking process.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>A unique record is shown for each version of a blocking chain, denoted by the <strong>Version</strong> column. A new record is created each time the blocking chain changes, which means that a blocked SPID was either added or removed from the chain between polling intervals. For some blocks this may happen frequently causing multiple records to be created, while others may not change at all for the duration of the block.</td>
</tr>
</tbody>
</table>

**Note:** The polling interval for blocks is determined by the Event Monitor **Polling Interval** (Navigator pane → Monitoring Service node → Settings → Event Monitor tab).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server</td>
<td>Name of the SQL Server hosting the SPID.</td>
</tr>
<tr>
<td>Start Time</td>
<td>Start time of the request.</td>
</tr>
<tr>
<td>Duration</td>
<td>The length of time that the block existed.</td>
</tr>
<tr>
<td>Plan</td>
<td>Use the <strong>Plan button</strong> to open a Plan Explorer session for the associated query plan.</td>
</tr>
<tr>
<td>Statement</td>
<td>The command text associated with the request.</td>
</tr>
<tr>
<td>Object</td>
<td>The object associated with the request.</td>
</tr>
<tr>
<td>Wait Time</td>
<td>Duration of wait time in milliseconds.</td>
</tr>
<tr>
<td>Wait Type</td>
<td>Name of the Wait Type. For more information about wait types see this MSDN article.</td>
</tr>
<tr>
<td>Wait Resource</td>
<td>Name of the resource on which the request is currently waiting.</td>
</tr>
<tr>
<td>Host</td>
<td>The client workstation specific to this session.</td>
</tr>
<tr>
<td>Application</td>
<td>The associated application.</td>
</tr>
<tr>
<td>Database</td>
<td>The associated database.</td>
</tr>
<tr>
<td>Login</td>
<td>The Login Name associated with the session.</td>
</tr>
<tr>
<td>Last Batch</td>
<td>The last time a client process completed batch execution.</td>
</tr>
<tr>
<td>Host Process ID</td>
<td>The process ID of the client program which initiated the session.</td>
</tr>
</tbody>
</table>
The Deadlocks tab displays information about deadlocks happening within your environment. Use the Deadlocks tab to identify and correct deadlocks on your monitored servers.

For a quick overview of the Deadlocks tab be sure to see the SQL Server Deadlocks video on the SentryOne TV website.

DEADLOCK COLLECTION

Deadlock collection is controlled through the Deadlock Source Settings. The Collect Deadlock Events setting is True by default, so whenever a new SQL Server Instance is watched, deadlock events will be automatically collected for that Instance. You may adjust deadlock collection as desired through the Settings pane (View menu → Settings).

Expand for Deadlock Settings Examples

If you wanted to disable deadlock collection Globally:

1. Select the Shared Groups node in the Navigator Pane (Global context).
2. In the Settings pane, use the top drop-down list and select SQL Server Settings.
3. Use the second drop-down list to select Deadlocks Source. You should now see the Deadlocks Source Settings that are being applied Globally.
4. Change the Collect Deadlock Events: Collection Setting to False, it will be saved automatically.

If you wanted to disable deadlock collection for an individual Instance:

1. Select the desired Instance node in the Navigator pane.
2. In the Settings pane, use the drop-down list and select Deadlocks Source.
3. You should now see the Deadlock Source Settings which are configured for the Instance. Change the Inherit From Parent Setting to False.
4. Change the Collect Deadlock Events: Collection Setting to False, it will be saved automatically.

DEADLOCK RETENTION

By default, Performance Analysis Blocking, Top SQL, and Deadlock data is retained for 15 days. This is controlled with the Keep Performance History For Setting found on the Storage tab of the Global Settings (Navigator pane → Configuration → Global Settings → Storage tab). For more information about retention see the Data Capacity Planning KB article.

DEADLOCK ALERTING

You may configure Actions in response to two SQL Server Deadlock Conditions. The SQL Server: Deadlock Condition occurs each time there is a deadlock in your environment. There are a number of Actions which you can take in response to this Condition occurring.

You may also configure an Action in response to the SQL Server Deadlock: Output Content Match (OCM) Condition. The OCM Condition can be useful in narrowing the notifications you see concerning deadlocks. The OCM Condition uses the entire deadlock XML to match against, making it a powerful tool in pinpointing specific deadlocks.
The Deadlocks tab has two modes, **Real Time** and **History**. The active mode is controlled through the toolbar. When the auto-refresh Play button is selected, the tab enters Real Time mode. In Real Time mode, any deadlocks which happened within the last 10 minutes will be displayed.

When the auto-refresh Pause button is selected, the tab goes into History mode. Select a time range and then use the Go toolbar button to view deadlocks which happened during that time period. Alternatively, use the Jump To Last Deadlock button to navigate to the last deadlock.

**FILTER**

The **top pane** provides client filters for specifying which records to display for the selected interval.

*Note:* The Filter pane is for Client side filtering only and does not impact what is actually collected. See the Deadlock Collection section of this topic for collection related information.

After changing any of the filters, it is necessary to click the Refresh button, or press F5, to apply the filter. To select more than one item in a list, use **shift-click** or **ctrl-click**. The available filters are:

- SQL Servers
- Show Top - depending on the length of the displayed interval, you may have a lot of data to work with. This setting controls how many of the rows are actually retrieved.

**DEADLOCK GRIDVIEW**

The **Deadlock Gridview** lists the records matching the current filters. The default filter is to display all records. Each record is expandable to view the details regarding the processes that were involved in the deadlock. For general background information about what constitutes a SQL Server deadlock, see this group of MSDN articles. For general information about the metrics displayed see the **Deadlock Metrics** section at the end of this topic.

**DEADLOCK GRAPH**

The **Deadlock Graph** is formed according to the captured deadlock XML. Process and Resource nodes are independently represented, along with any relationships which exist between them.

- The deadlock victim is highlighted with a red background.
- The numbers (1) and associated arrows, between each node, indicate the actual sequence of events which took place, leading to the deadlock.
- The letters indicate the requested lock mode.
- Selecting object nodes in the graphical representation will set focus to that object in the gridview.
- You may zoom in and out on the Deadlock Graph using **ctrl + mouse wheel** or through the right-click context menu.

**ADDITIONAL OPTIONS**

There are additional options available through the right-click context menu, including:

- Jump to Calendar - go to that event on the Event Calendar (requires server to be watched with EM).
• Jump to Top SQL - open the Top SQL tab highlighting the query involved in the deadlock event.
• Export Deadlock to XML - exports the deadlock in XML format.
• The divider between each of these panes can be dragged to resize or hide each pane

💡 You may open exported Deadlock XML files (*.xdl) in the SentryOne Client (File menu -> Open). Whenever you open a deadlock, a new tab displaying the deadlock information, including the deadlock graph will be created.

### DEADLOCK METRICS

#### Deadlock Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server</td>
<td>The SQL Server where the deadlock took place</td>
</tr>
<tr>
<td>Time</td>
<td>The time the deadlock occurred.</td>
</tr>
<tr>
<td>Victim SPID</td>
<td>The session process ID of the Victim involved in the deadlock.</td>
</tr>
<tr>
<td>Victim Host</td>
<td>The workstation belonging to the victim thread.</td>
</tr>
<tr>
<td>Victim Application</td>
<td>The application name belonging to the victim thread.</td>
</tr>
<tr>
<td>Victim Database</td>
<td>The name of the database on which the process took place</td>
</tr>
<tr>
<td>Victim Text Data</td>
<td>The associated text data of the victim.</td>
</tr>
<tr>
<td>Deadlock XML</td>
<td>This is the actual captured Deadlock XML.</td>
</tr>
</tbody>
</table>

#### Lock Details

The Lock Details area breaks down the deadlock by specific lock types, including the owners and waiters involved in each lock.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID [ecid]</td>
<td>The session process ID of the associated owner/waiter.</td>
</tr>
<tr>
<td>Plan</td>
<td>Use the Plan button to open a Plan Explorer session for the associated query plan.</td>
</tr>
<tr>
<td>Host</td>
<td>The workstation name.</td>
</tr>
<tr>
<td>Application</td>
<td>The associated application</td>
</tr>
<tr>
<td>Database</td>
<td>The associated database</td>
</tr>
<tr>
<td>Log Used</td>
<td>The amount of Log Space used by the process.</td>
</tr>
<tr>
<td>Deadlock Priority</td>
<td>Specifies the Deadlock Priority. Zero (0) or NORMAL is the default priority. In cases where each session has the same deadlock priority, SQL Server will choose the victim based on the least expensive session to roll back. For general information about the DEADLOCK_PRIORITY option see this MSDN article.</td>
</tr>
<tr>
<td>Wait Time</td>
<td>Time in (ms) milliseconds spent waiting on the resource.</td>
</tr>
<tr>
<td>Transaction Start Time</td>
<td>Time that the transaction began.</td>
</tr>
<tr>
<td>Last Batch Start Time</td>
<td>The last time a client process started batch execution.</td>
</tr>
<tr>
<td>Last Batch Completion Time</td>
<td>The last time a client process completed batch execution.</td>
</tr>
<tr>
<td>Mode/Type</td>
<td>The Mode/Type designates the resource lock mode. For general information on Lock Modes see this MSDN article.</td>
</tr>
</tbody>
</table>
### Status
State of the task.

### Isolation Level
The current transaction isolation level. For general information on isolation levels see [this MSDN article](https://docs.microsoft.com/en-us/sql/relational-databases/transaction-processing/services/isolation-levels?view=sql-server-ver15).

### Login Name
The Login Name associated with the session.

---

#### Owner/Waiter Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>Indicates the associated object name.</td>
</tr>
<tr>
<td>Line Number</td>
<td>The line number which was being executed when the lock occurred.</td>
</tr>
<tr>
<td>Text Data</td>
<td>The associated text data.</td>
</tr>
</tbody>
</table>

---

## 9.12 Quick Traces

### Applies to: SQL Sentry

**QUICK TRACE™**

A Quick Trace™ is a comprehensive snapshot of activity created by combining process-level data and trace events collected during a brief sample period. Various metrics such as CPU, I/O, recompiles, cache misses, cursor operations, etc., are automatically aggregated, and can be easily grouped and sorted, to provide a clear picture the processes, hosts, applications, or users responsible for activity during the sample.

Whereas the continuous Top SQL trace uses filters to collect only the heaviest events by default in order to maintain low overhead, a Quick Trace is not filtered and so collects *all* events. This is why Quick Trace is both time- and row-limited, to avoid impacting the performance of the target SQL Server.

A Quick Trace is typically run manually from the dashboard in response to observing high utilization in one or more metrics. For example, if you see an spike in Transactions/sec on the SQL Server Activity, you can run a Quick Trace, then sort by Transactions to determine the cause.

### COLUMN SET

There are four Column Sets that can be selected and modified to provide a pre-determined view of the data with preset columns and sorting. If a Quick Trace is launched by right clicking the Network, CPU, SQL Server Activity, or Disk IO charts, the appropriate column set will be used by default. The active column set can be easily changed by right-clicking the master or detail column header row and selecting the Column Chooser menu item. When you initiate a Quick Trace from the Quick Traces tab the SQL Activity Column Set will be used.

The Quick Traces tab lists all Quick Traces occurring during the active date range. Quick Traces can be executed manually or automatically via the Run Quick Trace action in response to a condition such as Runtime Threshold Max or Performance Counter Threshold Max.
RUNNING A QUICK TRACE

When Run Quick Trace is selected from the Quick Traces tab or from the right-click menu on the other applicable tabs, the options menu opens. In the first dialogue box, the user can enter the length of time the Quick Trace will run. In the second dialogue box the user can either Collect statement events or not by using the check-box. In the third box, the number of Trace Data rows can be limited to the desired amount.

Default Settings:
- The Quick Trace will run for 15 seconds.
- Collect statement events is unselected.
- Limit Trace Data to 10000 rows.

* Generally these settings are fine but may need to be adjusted in certain situations.

RESTRICTIONS

In order to avoid impacting server performance on very busy systems, there are certain cases where SentryOne will restrict Quick Trace functionality. See the table below for more information about these cases.

Quick Trace Restrictions

Case

A Quick Trace will be not allowed if a 100Mb adapter is present, and the last sample indicates there are >300 users or >3000 transactions per second for the targeted server.

This state is also checked before automated Quick Traces are run. If the state is detected, the Quick Trace is disallowed.

A Quick Trace is not recommended in the case where 1000Mb or above adapters are present, and the last sample indicates there are >500 users or >5000 transactions per second for the targeted server.

If this state is detected during a manually initiated Quick Trace, a warning will be generated indicating that the Quick Trace is not recommended. The state is also checked before automated Quick Traces are run. If the state is detected, the Quick Trace is disallowed.

⚠️ WARNING: SentryOne Version 7 introduced the above restrictions for Quick Trace functionality. For prior versions, use the above restrictions as a guideline when running A Quick Trace. Failure to follow these guidelines may cause a temporary suppression of transaction throughput. The risk of impact to performance is much greater if the network speed between the SentryOne Client and the target server is less than 1000Mbps.

EXPORTING A QUICK TRACE

Once a Quick Trace is complete, it can be opened for viewing. This can be done by double-clicking on it or by right-clicking and selecting open. From there, the file can be exported using the Export option in the File menu. There are two formats it can be exported in:
- Web Pages (.htm, .html)
- Microsoft Excel (.xls)

**DISPLAY**

The top pane provides a series of filters and controls for specifying which records to display for the specified interval. The available filters are

- SQL Servers
- Owner
- Source

The bottom pane lists the records matching the current filters. The default filter is to display all records.

**CONTROLS**

- **Top Pane**
  - Set filter - After changing any of the filters, it is necessary to click the Refresh button, or press F5, to apply the filter.
  - List box filters
    - Click on a combination of items in any of the lists to set the filters. The default filter is to display all records.
    - To select more than one item in a list, use shift-click or ctrl-click.
  - SQL Servers (dropdown) - Select server instance to run a Quick Trace.
  - Run Quick Trace - Start a Quick Trace against the select SQL Server instance.

- **Bottom Pane**
  - Sort Column - click any column header to sort by that column.
  - Group By box - drag any column header here to group by that column.
  - Double click on a row - View the selected Quick Trace.
  - Context Items
    - Jump to Dashboard - opens the Dashboard with the selected QuickTrace event overlaid on each of the charts.
    - Open - View the selected Quick Trace

**VIEWER**

The Quick Trace Viewer presents a set of nested controls to inspect the trace. These controls behave the same as on the Quick Traces tab, and will not be covered here.

Right clicking on a trace record presents a context menu that will allow you to copy the event row data to the clipboard or kill the process associated with that trace event. **Note:** the individual text can be copied from the Text Data column, once highlighted, using (Ctrl + C) or the context menu.

Additional controls presented include:

- Reload - Run a new Quick Trace of the same target.
- Refresh - Available when a Quick Trace is run in real-time, and causes the Quick Trace to execute with same parameters.
• Options - Opens the Options pane. Available options in the Viewer are:
  • Auto-highlight detail rows when sorting by an event totals column
  • Freeze metadata columns
  • Hide rows with empty values in sort column
  • Maintain row position when changing sort
  • Show Quick Trace process
  • Show statement events if available
  • Show system processes
• Close - Close the Quick Trace Viewer

9.13 Performance Analysis Overview

The Performance Analysis Overview provides the user with a summary of performance data for all watched Instances within a group. To open the Performance Analysis Overview, use the right-click context menu of any group node. Selecting the All Targets node will open an overview showing all of the Instances that are being watched. Selecting a custom made group node will only display the Instances within that group.

• Use the Show Avg Over drop-down list box to change the interval for the summarized data.
• Use the Show drop-down list box to view either Windows, SQL Servers, or Analysis Services.
• Double-click on any Target or Instance to open Performance Analysis for that Instance.

Clicking on a performance metric will cause a historical graph to appear at the bottom of the screen.

Note: Historical graphs are not available for all metrics.

Please see the Performance Metrics section of the User Guide for a detailed description of the various performance metrics.

The default setting for the Performance Analysis Overview Skin is black, but can be set to standard via the User Preferences window. (Tools->User Preferences->Performance Analysis)

FAILOVER SIMULATION

When the monitored Instances include AlwaysOn Availability Groups, the user has the ability to simulate what the performance values would be in the event of a failover. To do this, expand the nodes until the Availability Groups level has been reached. Right-clicking on an Availability Group that is currently the Primary will produce a context menu that gives the user the option to select which Secondary replica that they would like to simulate a failover to. The user can also simulate failover values from a Secondary by right-clicking on the Secondary replica and selecting Simulate Failover from the context menu. To stop the failover simulation, select Reset Simulation from the context menu of an Availability Group. The simulation can also be reset from the Tools menu.

Once the user has selected to simulate the failover, simulated values will appear as italicized.
Clicking on a simulated value will cause a historical graph of the actual values to appear along with a Simulated Values chart to allow the user to quickly compare the simulated value to the actual value.

**FILTERS**

In the top right corner of each filter column header is a filter button. Once selected, the user can choose the operator and value to filter the data. Once a filter is set, it will appear along the bottom panel of the window. Edits to the filter can be made here by simply clicking ‘Edit Filter’ and changing the criteria. It can also be disabled by clearing the checkbox or closed completely using the ‘X’ on the left side.

**COLOR INDICATORS**

Instance names will appear in different colors depending on the status of the Instance.

<table>
<thead>
<tr>
<th>Color</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray</td>
<td>Normal</td>
</tr>
<tr>
<td><em>(Black in the Standard View)</em></td>
<td></td>
</tr>
<tr>
<td>Orange</td>
<td>A metric has crossed the Warning threshold at or below this level.</td>
</tr>
<tr>
<td>Red</td>
<td>A metric has crossed the Critical threshold at or below this level.</td>
</tr>
<tr>
<td>Black</td>
<td>Instance Offline</td>
</tr>
<tr>
<td><em>(Gray in the Standard View)</em></td>
<td></td>
</tr>
</tbody>
</table>

**PERFORMANCE ANALYSIS OVERVIEW THRESHOLDS**

<table>
<thead>
<tr>
<th>Target Level</th>
<th>Normal</th>
<th>Warning</th>
<th>Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Outbound Queue</td>
<td>&gt;=3</td>
<td>&gt;=3 for 20 second duration</td>
<td></td>
</tr>
<tr>
<td>Network Inbound Errors</td>
<td>&gt;0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Outbound Errors</td>
<td>&gt;0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory Free (MB)</td>
<td>&lt;100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Context Switches</td>
<td>&lt;5000 times # of cores</td>
<td>&gt;=5000 times # of cores</td>
<td>&gt;=7500 times # of cores</td>
</tr>
<tr>
<td>CPU Processing Time %</td>
<td>85-89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Queue Length</td>
<td>&gt;2 times the # of cores</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target and SQL Server Level</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk Reads ms/IO</td>
<td>&lt;20</td>
<td>20-29</td>
<td>&gt;=30</td>
</tr>
<tr>
<td>Disk Writes ms/IO</td>
<td>&lt;20</td>
<td>20-34</td>
<td>&gt;=35</td>
</tr>
</tbody>
</table>
### Target Level

<table>
<thead>
<tr>
<th>SQL Server Activity Blocks</th>
<th>Normal</th>
<th>Warning</th>
<th>Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&gt;0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQL Server Activity Deadlocks</th>
<th>&gt;0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Average Wait Time (ms) CPU %</th>
<th>&gt;25</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SQL Server Memory Grants Pending</th>
<th>&gt;2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SQL Server Memory PLE</th>
<th>&lt;600</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Disk IO Lazy Writes</th>
<th>&gt;20</th>
</tr>
</thead>
</table>

### SSAS Level

<table>
<thead>
<tr>
<th>SSAS Formula Engine Queued Jobs</th>
<th>&gt;0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSAS Storage Engine Queued Jobs</th>
<th>&gt;0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSAS Memory Cache Evictions</th>
<th>&gt;0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSAS Storage Temp File (KB)</th>
<th>&gt;0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SSAS Memory (KB) Shrunk</th>
<th>&gt;0</th>
</tr>
</thead>
</table>

### Performance Analysis Toolbar

The Performance Analysis tabs can be synchronized to view performance data for a historical period, or in real time mode for the previous 10 minutes. This is coordinated using the toolbar. See the chart below for more information about the toolbar.

<table>
<thead>
<tr>
<th>#</th>
<th>Button Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Print Preview</td>
<td>Print preview not active in Performance Analysis.</td>
</tr>
<tr>
<td>2</td>
<td>Print</td>
<td>Print function not active in Performance Analysis.</td>
</tr>
<tr>
<td>3</td>
<td>New Plan Explorer Session</td>
<td>Opens a new Plan Explorer Session.</td>
</tr>
<tr>
<td>4</td>
<td>Open Plan File</td>
<td>Open execution plan file.</td>
</tr>
<tr>
<td>5</td>
<td>Save</td>
<td>Save execution plan file.</td>
</tr>
<tr>
<td>6</td>
<td>Start Date</td>
<td>Set the beginning date for viewing a range of performance related data.</td>
</tr>
<tr>
<td>7</td>
<td>Start Time</td>
<td>Set the beginning time for viewing a range of performance related data</td>
</tr>
<tr>
<td>8</td>
<td>End Date</td>
<td>Set the end date for viewing a range of performance related data.</td>
</tr>
<tr>
<td>#</td>
<td>Button Name</td>
<td>Description</td>
</tr>
<tr>
<td>----</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>9</td>
<td>End Time</td>
<td>Set the end time for viewing a range of performance related data.</td>
</tr>
<tr>
<td>10</td>
<td>Go</td>
<td>Activate the selected start/end time range.</td>
</tr>
<tr>
<td>11</td>
<td>Go To Current Time</td>
<td>Move forward so that the display range ends at the current time.</td>
</tr>
<tr>
<td>12</td>
<td>Zoom Out</td>
<td>Increase the date range.</td>
</tr>
<tr>
<td>13</td>
<td>Zoom In</td>
<td>Decrease the date range.</td>
</tr>
<tr>
<td>14</td>
<td>Previous Interval</td>
<td>Skip back in time one full interval.</td>
</tr>
<tr>
<td>15</td>
<td>Previous Small Interval</td>
<td>Skip back in time one small unit of the interval scale.</td>
</tr>
<tr>
<td>16</td>
<td>Next Small Interval</td>
<td>Skip forward in time one small unit of the interval scale.</td>
</tr>
<tr>
<td>17</td>
<td>Next Interval</td>
<td>Skip forward in time a full interval.</td>
</tr>
<tr>
<td>18</td>
<td>Synchronize Tabs</td>
<td>Automatically sets all tabs to the active date range (enabled by default).</td>
</tr>
<tr>
<td>19</td>
<td>Start Auto-refresh</td>
<td>Enables automatic updating of real time data for the current tab, or all tabs if Synchronize Tabs is enabled.</td>
</tr>
<tr>
<td>20</td>
<td>Pause Auto-refresh</td>
<td>Pauses automatic updating of real time data for the current tab, or all tabs if Synchronize Tabs is enabled.</td>
</tr>
<tr>
<td>21</td>
<td>Refresh</td>
<td>Refresh the active view by retrieving the latest data for the active date range.</td>
</tr>
<tr>
<td>22</td>
<td>Show History</td>
<td>Toggle between history and sample modes when on the dashboard.</td>
</tr>
<tr>
<td>23</td>
<td>Run Quick Trace</td>
<td>Used to execute a Quick Trace.</td>
</tr>
<tr>
<td>24</td>
<td>Show Windows Processes</td>
<td>Opens the current Windows processes for the server.</td>
</tr>
<tr>
<td>25</td>
<td>Jump to Calendar</td>
<td>Opens the Calendar tab in the workspace area.</td>
</tr>
<tr>
<td>26</td>
<td>Show Advisory Condition Markers</td>
<td>Enables and disables Advisory Condition glyphs and Event Markers</td>
</tr>
</tbody>
</table>

9.15 Data Grids

DATA GRID OVERVIEW

On the different tabs in SentryOne, various types of grids are used to display data. All of the grids contain applicable information according to which tab they are on. These grids are very easy to customize and provide the user with a variety of organizational methods, with a few simple clicks.

GROUP BY

Many of the grids found in SentryOne can be grouped by column fields. This can be done by dragging the column to the title area of the grid. This will group all of the data into the respective categories of the column selected. The Group By function can also be done by right-clicking on the column header and selecting the option from the menu.

SORTING

The grids that are shown in SentryOne can be easily sorted to organize data. This can be done by clicking the column header which will automatically sort according to the values in that column. Clicking the same column again will toggle between ascending and descending order for the data in that field. This can also be done using the right-click menu that opens when the column header is selected.

FILTERS

On many of the fields found on grids in SentryOne, a filter can be used along with different sorting methods to organize data. The filters show up on applicable column headers on the top-right side of the cell when the mouse is over it. The filter can be selected by clicking the small filter button, which will bring up the filter menu options. The Filter Editor is available from the right-click menu also. This feature allows the user to specify filter conditions by typing them into the
window that opens.

**EXPORTING GRIDS**

The Data Grids found in SentryOne provide a great deal of information to the user in a form that is very easy to customize. Once the desired organization of the columns has been achieved, the user can export the information in a variety of formats. This can be done using the file menu on the applicable tabs of the SentryOne Client. Select Export Data, give a file name, then select the desired file type.

**Types of Files:**

- CSV Files (*.csv)
- Microsoft Excel (*.xls)
- Adobe Acrobat Files (*.pdf)
- Web Pages (*.htm; *.html)

**9.16 Virtual Machine Support**

**HYPER-V SUPPORT**

After monitoring a Hyper-V Guest target with SentryOne, the Host of that target will appear under the Virtualization node in the Navigator Pane.

When a Hyper-V Host target is monitored with Win Sentry, the Host and all of the Host’s Guests will appear under the Virtualization node in the Navigator Pane. Additionally, memory ballooning and vCPU information will now be presented on the dashboards on Hyper-V Guest targets. The Host information will be provided on the Dashboard as well.

**VMWARE SUPPORT**

To enable VMware support, a vCenter must be added. To add a vCenter, right-click on the VMware node under the Virtualization node in the Navigator pane and select the Add vCenter option.

Once the vCenter has been added, any VMware Guest targets will now display CPU Ready Time, Co-Stop, and memory ballooning metrics on their dashboards.

**NOTE:** When adding a vCenter to a site with multiple Monitoring Services, in order to avoid potential overhead associated with Windows session authentication, it is recommended that you use a local vSphere account.

**VIRTUAL MACHINE PERFORMANCE COUNTERS**

In addition to getting VM-related metrics on the Performance Dashboards, Virtual Machine Performance Counters are also available for use in Advisory Conditions and are available for viewing in the Performance Counter History Report.

**NOTE:** Memory ballooning information is only available for Guests with dynamic memory enabled. vCPU information will only be displayed on the Sample Mode Dashboard CPU charts if the value is > 1%.
Cloud.sentryone.com is a web service hosted by SentryOne which allows users to upload and share certain information about their monitored SentryOne environment. Once the cloud upload is configured, a partial replica of your on-premises SentryOne database will begin syncing to the SentryOne cloud.

During the configuration process, you will first register a user with cloud.sentryone.com, and then enable which monitored servers in your environment are synced to the cloud. Initially, only the user who completed the configuration process will have access to the synced information on cloud.sentryone.com. There are several options for sharing your database outlined here, including the ability to share with individuals, and the ability to define a subset of which servers will be visible to those users with which you share your database.

**WHAT'S AVAILABLE IN THE CLOUD?**

The cloud.sentryone.com offering includes the following features:

- **Server Health Status** - A high level view of the health of your environment
- **Server Details** - A detailed view of your environment's performance, including reports and an area to manage cloud sharing

**CLOUD CONFIGURATION IN THE SENTRYONE CLIENT**

To configure your environment for cloud synchronization, expand the Monitoring Service node in the Navigator Pane and open the cloud.sentryone.com node.

**REGISTRATION**

From the cloud.sentryone.com tab, click the Configure Sync button to begin the registration process. If you have an existing SQLPerformance.com account, it can be used for cloud.sentryone.com. If you do not have an account, you may create one using the right side of the registration form.

**CONFIGURATION**

After completing the registration process and logging in, you will see a list of all monitored Targets and Instances in your environment. You will also notice that the initial state of the synchronization is off.

**CHOOSING WHICH TARGETS AND INSTANCES TO SYNC**

From the list of Targets and Instances in your environment, you may choose which ones you would like to enable the cloud synchronization process. If you would like to select all Targets and Instances, you can simply right-click and select Enable -> All from the context menu.

**CHOOSING WHAT TO SYNC**

When an Instance is selected to be synced with the cloud, Top SQL, Blocking SQL, and Deadlock information are enabled for sync by default. With the realization that you may have sensitive data in your environment, you have control over which types of information get synced to the cloud. You can disable any of these options by drilling into the Instance and unchecking the box next to the option that you would like to disable.
**ALIASING TARGETS, INSTANCES, AND DATABASES**

When you synchronize your SentryOne data with the cloud you have complete control over who sees this data and how you choose to share it. As an added security feature, you may choose to alias the names of Targets, servers, and databases before they are synchronized. This obfuscation allows you to hide potentially sensitive data.

You have the ability to add aliases by typing in the Alias field, or you can auto-generate aliases through the context menu. Once you have made your selections, toggle the Off button to the On position, and after a brief initialization process, synchronization will begin. Synchronization will continue to occur at the Sync Frequency that is selected.

⚠️ **NOTE:** Changes in alias names or objects enabled for syncing require a save before the settings will take effect.

**REstricting Visibility of servers**

- SentryOne supports restricting server visibility in the cloud through the application of Rights Based Security. With Rights Based Security, a user can be assigned a limited set of visible Sites, Target Groups, or Instances. Those rights that you configure in the SentryOne Client for a user will be honored in the cloud.

- To restrict what a shared user has access to, first create a user in the SentryOne client and restrict their rights via Rights Based Security. That user will then be available in the Link to SentryOne Client User drop down while sharing.

⚠️ **NOTE:** When creating a user for cloud sharing purposes, it is not necessary for the user to have a SQL Server login.

### 9.17.1 Server Details

The Server Details area of the cloud includes the following features:

**VIEWS**

- **Performance Overview:** The Performance Analysis Overview provides the user with a summary of performance data for all synced servers. This is the web equivalent to the SentryOne Client Performance Analysis Overview.

- **Events Log:** The Events Log provides the user with a listing of Advisory Condition Events for synced servers. Actual evaluation values are not provided within the cloud, only the opened and closed status of the Advisory Conditions. This is the web equivalent to the SentryOne Client Global Events Log.

- **Reports:** Expansive list of Reports for your monitored environment. This is the web equivalent to the SentryOne Client Reporting options.

**MANAGEMENT OPTIONS**

- **Sync History:** The complete sync history of your SentryOne repository database. This includes a listing of each table that has been synced with cloud.sentryone.com.

- **Database Details:** Provides summary information about your cloud installation. The sharing
of your cloud installation can be managed from this view.

**SHARING YOUR CLOUD DATABASE**

You can share your cloud database by accessing the Database Details page on cloud.sentryone.com. From the Member Section Overview page, select the Launch Database icon and then select the Database Details tab. From the Database Details tab, locate the **Database Sharing** section.

You can share your database with either an existing cloud.sentryone.com user or a non-member. You may also choose to restrict what servers are visible to those users with whom you share your database. For more information, see the sections below.

**SHARING WITH AN EXISTING USER:**

- If you would like to share your database with an **existing user**, you can do so either via email or cloud.sentryone.com username. The existing user will receive an email letting them know that you have shared a database with them, and the next time they log in, they will have access to the shared database.

**SHARING WITH A NEW USER:**

- If you would like to share your database with a **non-member**, simply enter their email address. They will be invited to register on cloud.sentryone.com via email, and once they have completed the registration process, they will have access to the shared database.

**RESTRICTING VISIBILITY OF SERVERS**

- SentryOne supports restricting server visibility in the cloud through the application of **Rights Based Security**. With Rights Based Security, a user can be assigned a limited set of visible Sites, Target Groups, or Instances. Those rights that you configure in the SentryOne Client for a user will be honored in the cloud.

- To restrict what a shared user has access to, first create a user in the SentryOne client and restrict their rights via **Rights Based Security**. That user will then be available in the **Link to SentryOne Client User** drop down while sharing.

> **NOTE:** When creating a user for cloud sharing purposes, it is not necessary for the user to have a SQL Server login.

### 9.17.2 Server Health Status

Server Health Status provides you with a high level view of the health of your environment.

**SERVER HEALTH STATUS AND ADVISORY CONDITIONS**

Some functionality of the Server Health Status is dependent on the use of Advisory Conditions. In addition to your own Advisory Conditions, you can download the latest Advisory Condition Pack from the Tools menu in the SentryOne client. This pack includes many Advisory Conditions that will assist in monitoring performance issues, such as high CPU usage, disk waits, and more.

**ELEMENTS OF SERVER HEALTH STATUS**
Views

There are three available views on the Server Health Status page. The Server Health Indicator and Score will display different information depending on what view you are using.

- **Wait Time / Session** – Uses a score based on wait stats
- **Events by Tag** – Health Indicator displays information relating to Advisory Condition tags
- **Events By Severity** – Health Indicator displays information relating to Advisory Condition severity

Health Indicator

The Health Indicator provides a visual representation of a server’s health.

In the Wait Time / Session view, the Indicator is divided into 5 sections. Going clockwise, these sections represent waits of the following categories: CPU, Memory, Disk, Other, and Network. As these waits increase, the corresponding section will begin to fill and the Health Indicator will begin to change color, going from green to yellow to red.

In the Events by Tag view, each section represents the tags that are applied to the Advisory Conditions in your environment.

In the Events by Severity view, the Health Indicator is divided into quadrants representing Low, Medium, High, and Critical.

Score

Scores give you a metric to use to quickly evaluate your server’s health.

Wait Time / Session relies on wait stats data that are collected by SentryOne. These wait stats are broken down by major resource category (Network, CPU, Memory, Disk, Other). The "Other" category is for a few other important wait types that can either affect performance in more than one major category, or cannot be directly attributed to any category with absolute certainty, such as backups and parallelism respectively. Each category is calculated independently. The displayed value for a category is calculated as the waits for the category divided by the user sessions. The values for waits and user sessions are based on the most recent values present in the selected timeframe. Each value is then rounded and displayed as the category’s value. The overall score is the sum of the five rounded categories.

The Events by Tag and Events by Severity views work by incorporating Advisory Conditions. Both user-defined and prepackaged Advisory Conditions are used to calculate the scores. The score for these views is based on the severity of all open Advisory Condition events within the selected time period. The severity points are as follows:

- Low - 1 Point
- Medium - 3 Points
- High - 6 Points
- Critical - 9 Points

The weight of an open event diminishes with its age. The score being presented is 100 minus the sum of the weighted value of open events.

NOTE: Closed events are not factored into the Server Health score. Events can be closed in the Events Log in the SentryOne client.

Uptime Chart

The uptime chart will indicate any downtime that occurred within the defined timeframe.
Downtime will be displayed as a red gap within the chart. Uptime percentage is displayed below the chart.

**Events Log**

When at the Instance level, you can view the Events Log. The Events Log displays any Advisory Condition events that were active during the time period that is selected. The time period can be adjusted by using the Time Period Filter in the top right corner.

**Navigation Bar**

- **Detail View** – This view displays more detailed information without the need to drill into a specific Target or Instance.
- **Summary View** – This view provides a minimalist view of your environment, allowing you to quickly spot servers that may be experiencing sub-optimal conditions.
- **Group Mode** – This mode is available in either view and allows you to see Targets/Instances as they are grouped in the SentryOne Client Navigator pane.
- **Zoom Out**
- **Zoom In**
- **Time Period Filter** – The Time Period Filter allows you to see the Server Health of your environment over the selected period of time.
The concepts and terminology covered in this section are important for a full understanding of SentryOne software.

Earlier versions of SentryOne focused primarily on SQL Server Agent jobs, however, this is no longer the case. Event Manager’s new Event Connector Architecture now enables monitoring and management of different types of scheduling systems, and various types of jobs, tasks and status events.

This necessitated some dramatic changes to both our system architecture and terminology. As a result, throughout this guide more generic terms such as "event store connection" and "event object" (or simply "connection" and "object") are used where before references to "SQL Servers" or "jobs" were used. The following topics cover these architectural and terminology changes in detail.

- Event Connector Architecture
- Execution Agents
- Execution Engines
- Event Stores
- Event Store Connections
- Event Providers
- Event Sources
  - SQL Server Agent Jobs
  - SQL Server Agent Alerts
  - Reporting Services Reports
  - SQL Server Agent Log
  - Windows Tasks
  - Windows Event Logs
- Event Objects
  - Active Event Objects
  - Status Events
  - Hybrid Event Objects
  - Event Instances

10.1 Event Connector Architecture

SentryOne’s event connector system encompasses the entire process by which SentryOne collects historical and active status information from various event stores, and translates the data into the calendars, lists, and graphs in the SentryOne Client. The data collected is also used to trigger the various actions carried out by the SentryOne Monitoring Service, such as sending email notifications.

In a nutshell, execution agents trigger execution engines which write data to event stores. There are many different types of agents, engines, and stores, and it’s important to note that this process isn’t new, and is not something we invented. However, SentryOne’s primary focus is collecting data from the respective stores, translating it, correlating it, rendering it in a meaningful fashion to
the DBA, and enabling automated actions based upon it. So we've had to define some terms which abstract these components to make the process by which SentryOne collects and uses this data more understandable. Creating new terms isn't something we do for fun, but rather something we've had to do here simply because all other existing "enterprise" schedulers utilize their own proprietary execution agents and engines. Since they don't "wrap" other agents and engines as we do, there probably wasn't much of a need for these terms before now.

As an aside, we believe our "agent-less" approach is superior to the "proprietary scheduling agent" approach employed by "enterprise" schedulers in many ways, including significantly reduced installation, configuration and maintenance overhead, as well as the ability to integrate with systems such as SQL Server Agent in a much more native fashion. For example, we don't work with SQL Agent jobs via the command line using osql.exe as other schedulers do - instead we talk directly to SQL Server in its native languages (DMO, SMO, ADO.NET, etc.). In general, we let existing scheduling agents do what they do best, schedule jobs and tasks, and we don't try to replace them.

The part of the process that is proprietary to SentryOne is how it uses event providers to access and translate the event store data over event store connections into the event objects used by the SentryOne Client and Server. Again, we aren't just dealing with SQL Agent jobs any more, so we've had to come up with a few terms to abstract these components to make them more understandable and manageable. It's also important to note that SentryOne is a 100% .NET-based application, so this object-oriented approach is very much inline with how we've designed the software, and in many ways these concepts and terms are descended from it.

10.2 Execution Agents

**Execution agents** represent any process responsible for the execution of one or more execution engines. There are two types of execution agents: schedulers and ad hoc.

**SCHEDULERS**

Schedulers of course trigger execution engines on an automated basis at predetermined times and/or intervals. There are many different types of scheduling agents – there are those like SQL Agent and Task Scheduler which can be used by anyone to schedule jobs and tasks. There are also “proprietary” schedulers. These are schedulers that exist as part of some application but don’t operate as standalone schedulers, and therefore can’t be used to schedule tasks unrelated to the application itself. Many anti-virus systems, disk defragmentation systems, and network backup systems fall into this category.

The schedulers that SentryOne is most interested in are those executing tasks that:

- can impact SQL Server performance.
- have dependency relationships with other SQL Server-related jobs or tasks.

Schedulers always have job (or task) history logs, which record data such as start time, end time, duration, and job output. If the Scheduler allows ad hoc (unscheduled) execution of its jobs, this will also normally be logged. Good examples of this are manually executed SQL Agent jobs and Windows Task Scheduler tasks which are written to the respective history logs just as scheduled executions are.
It’s important to note the SQL Server Agent isn’t only a scheduling agent - it has job subsystems that manage job steps and schedules, but it also contains an "Alert engine" which detects various conditions of the SQL Server process and performance counters, and it has the ability to auto-restart a failed SQL Server Agent or SQL Monitoring Service.

Schedulers sometimes have their own separate logs which record general information, errors or warning conditions not necessarily related to jobs. A good example of this is the SQL Server Agent log. Unlike SQL Server Agent, Windows Task Scheduler does not have its own log, only a task history log. In the case of SQL Server Agent, SentryOne collects data from both the SQL Server Agent Log itself as well as its job history log.

**AD HOC**

Non-scheduled executions are triggered by ad hoc agents, which include users and response-based systems.

When a user executes a job manually, it’s considered an ad hoc execution. Likewise when some other active process detects a transient condition in the environment and triggers an execution engine as a result it is considered a response-based ad hoc execution.

SentryOne is a great example of a response-based system - it is able to detect various conditions related to event sources and trigger a variety of pre-defined actions in response. See Alerting and Response System for more info.

In the case where a user executes an engine directly, there will be no scheduler logs, and SentryOne will be dependent on the engine’s own execution logs if they exist. If the engine does not log or support active status queries via API or other interface, then we are out of luck and SentryOne can’t do anything with it – we can’t show the event instances on a calendar, trigger conditions and actions, and/or monitor performance for the event. An example of this is an SSIS package that is run from within SSMS without package logging enabled, or a manually executed VBScript (.vbs) file that doesn’t perform any logging.

The most important point to remember regarding execution agents is that if the scheduler and/or the execution engine logs event data, and/or either allows active status queries, then SentryOne can work with it.

### 10.3 Execution Engines

An execution engine is just a fancy name for any application or program that does some type of work, typically background processing. Some examples are:

- SQL Agent jobs
- sqlmaint.exe – the executable responsible for SQL Agent Maintenance Plan functions
- ReportServer service – a Windows service which processes Reporting Services reports and produces associated output
- Task Scheduler – a Windows service responsible for running tasks
- Disk defragmentation software – an executable which defragments logical disk drives
- Anti-virus software – an executable which scans a system for viruses
One of the primary differences between execution engines and other programs is that they usually run as background processes, meaning they quietly go about their work without any significant user interface or user interaction. In other words, execution engines are not like UI-driven productivity applications like Microsoft Word or Excel. Seems simple enough, but it’s an important concept.

There’s another important characteristic of most execution engines – they throw off status information, typically in the form of an execution log. This element is key for many of the functions of SentryOne, including notifications, chaining, queuing, etc. If an execution engine does not log status information to its own execution log, then SentryOne will default to use the logs of the associated scheduler. The fact that it doesn’t have an execution log doesn’t mean that it’s not an execution engine, just that for SentryOne’s purposes there isn’t really much that can be done with it – we need some type of execution log data with an associated provider to show the event instance on the calendar, trigger conditions and actions, monitor performance, etc.

10.4 Event Stores

An event store is simply a collection of like event instances kept in a common location, typically a database or file-based storage. Some examples are shown in the table below:

<table>
<thead>
<tr>
<th>Event Provider</th>
<th>Provider Type</th>
<th>Event Type</th>
<th>Event Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Agent</td>
<td>Scheduler</td>
<td>Status</td>
<td>File Storage</td>
</tr>
<tr>
<td>SQL Server Agent Job</td>
<td>Execution Engine</td>
<td>Active</td>
<td>SQL Server database (MSDB)</td>
</tr>
<tr>
<td>Windows Task Scheduler</td>
<td>Scheduler</td>
<td>Active</td>
<td>Windows API + File Storage</td>
</tr>
<tr>
<td>Windows Event Log</td>
<td>Event Log</td>
<td>Status</td>
<td>Windows API</td>
</tr>
</tbody>
</table>

**EVENT LOGS**

Event logs are a type of event store which hold only status event instances. The Windows Event Logs, SQL Server Logs, and SQL Server Agent Logs are examples of event logs.

10.5 Event Store Connections

An event store connection (usually referred to simply as a connection) is used by event sources to access data in event stores via some network transport mechanism.

In the SentryOne Client, each SQL Server and Windows instance node represents a single connection.

For example, the Jobs event source above utilizes the “SQL Server Agent Job” event provider over an ADO.NET connection to the SQL Server to access event data stored in the MSDB database.

For maximum efficiency and performance, SentryOne allows a connection to be shared by multiple event sources simultaneously - all event sources under the SQL Server Instance node above (Jobs, Maintenance Plans, Alerts, Reporting Services, and SQL Agent Log) utilize the same physical network connection. This concept is illustrated in more detail in the Enterprise Architecture Diagram (‘Enterprise Architecture ’ in the on-line documentation).

**CONFIGURING INSTANCE-LEVEL SETTINGS**
Whenever you single-click on an Instance node in the Navigator, the Conditions and Settings pane will automatically refresh to display all conditions, actions and settings for the selected Instance. Any changes made here will apply to the selected Instance only, overriding any globally configured settings. Likewise, Instance-level settings can be overridden at the object level by selecting an object node in the Navigator.

10.6 Event Providers

An event provider is a module which enables SentryOne to communicate with and translate data from an event store over an event store connection into a common format which can be used by the various parts of the SentryOne system, including calendars, graphs, and notifications. A SentryOne event provider is analogous to an OLEDB provider or ODBC driver. There are three types of event providers: **Schedulers**, **Execution Engines**, and **Event Logs**.

**Scheduler** providers enable interfacing with the native scheduler logs, such as for SQL Server Agent or Windows Task Scheduler.

**Execution engine** providers connect with the native stores for execution engines for collecting historical information, and active status information if supported by the provider. Not all providers support active status queries.

**Event log** providers enable translation of status event data contained in a log-type stores, such as for the SQL Agent Alert log or the Windows Event Log.

SentryOne ships with several providers, including:

<table>
<thead>
<tr>
<th>Event Provider</th>
<th>Provider Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Agent</td>
<td>Scheduler</td>
</tr>
<tr>
<td>SQL Server Agent Job</td>
<td>Execution Engine</td>
</tr>
<tr>
<td>SQL Server Agent Alert</td>
<td>Event Log</td>
</tr>
<tr>
<td>Maintenance Plan</td>
<td>Execution Engine</td>
</tr>
<tr>
<td>Reporting Services</td>
<td>Execution Engine</td>
</tr>
<tr>
<td>Window Task Scheduler</td>
<td>Scheduler</td>
</tr>
</tbody>
</table>

SentryOne utilizes a “plug-in” architecture which allows future modules to be added on, typically without requiring install of a new version of the SentryOne software.

10.7 Event Sources
An **event source** represents a unique instance of an **event provider** operating over a single **event store connection**. In SentryOne's **Navigator pane**, every sub-node underneath a SQL Server node represents a unique event source.

Below is a current listing of all event sources currently supported by SentryOne, along with its associated event objects, providers and Instances:

<table>
<thead>
<tr>
<th>Item</th>
<th>Navigator Node</th>
<th>Event Provider</th>
<th>Instance Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Agent Jobs</td>
<td>Jobs</td>
<td>SQL Server Agent Job</td>
<td>SQL Server</td>
</tr>
<tr>
<td>SQL Server Agent Alerts</td>
<td>Alerts</td>
<td>SQL Server Agent Alert</td>
<td>SQL Server</td>
</tr>
<tr>
<td>SQL Server Agent Log</td>
<td>SQL Server Agent Log</td>
<td>SQL Server Agent Log</td>
<td>SQL Server</td>
</tr>
<tr>
<td>Maintenance Plans</td>
<td>Maintenance Plans</td>
<td>Maintenance Plan</td>
<td>SQL Server</td>
</tr>
<tr>
<td>Reporting Services Reports</td>
<td>Reporting Services</td>
<td>Reporting Services</td>
<td>SQL Server</td>
</tr>
<tr>
<td>Windows Tasks</td>
<td>Tasks</td>
<td>Window Task Scheduler</td>
<td>Windows Instance</td>
</tr>
<tr>
<td>Windows Event Logs</td>
<td>Windows Event Logs</td>
<td>Windows Event Log</td>
<td>Windows Instance</td>
</tr>
</tbody>
</table>

10.7.1 SQL Server Agent Jobs

SQL Server Agent Jobs are listed for each SQL Server Instance under the Jobs node. Expand the node to view the list of jobs, or single-click the node to view a calendar for all jobs. The SQL Server Agent Job **event provider** is used behind the scenes to access job metadata, history, and active status information.

Using the SentryOne Client, you can perform all of the same tasks related to jobs that you would previously do via SSMS, plus many more features such as visual scheduling, performance monitoring, reporting, queuing and chaining.

10.7.2 SQL Server Agent Alerts

**SQL SERVER AGENT ALERTS**

SQL Server Agent Alerts are listed for each SQL Server Instance under the Alerts node. Expand the node to view the list of alerts, or single-click the node to view a calendar for all alerts. The SQL Server Agent Alert **event provider** is used behind the scenes to access alert metadata and history information.

Out of the box SentryOne supports SQL Server Agent event alerts, but SentryOne can be configured to alert you on SQL Server Agent WMI event alerts. Additionally, SentryOne can alert you about general performance conditions, please see the Setting up Automated Performance Alerting KB article.
NOTE: In order to enable server alert collection and notification without the use of agents on each SQL Server, SentryOne must install one simple job (SentryOne Alert Trap) and stored procedure (msdb.dbo.spTrapAlert_20), and one small table (msdb.dbo.SQL_Sentry_AlertLog_20) on each watched server.

SentryOne must also make a minor configuration change to each server alert in order to enable alert collection and notification - the “Execute job” setting is updated to use the “SentryOne Alert Trap” job. If an alert is already set to execute another job, the “Execute job” settings for that alert will not be updated, and SentryOne will not be able to generate notifications for the alert.

One of the most common reasons the “Execute job” setting may already be set to a job is for logging and/or notification purposes, in which case SentryOne may be able to safely assume control of this function. Some of the advantages of using SentryOne instead are that its logging and notification processes are centralized (versus distributed on each server), they are not MAPI-dependent, and they can easily be made redundant by using more than one SentryOne Monitoring Service.

A complete list of objects SentryOne places on a watched server can be found here.

NOTE: In order to watch alerts on SQL Server 2005 and above Instances, token replacement must be enabled for the SQL Server Agent. This is disabled by default as a security precaution. More information on this setting can be found in SQL Server Books Online. You can also configure SentryOne to auto-enable SQL Agent tokens at the Global and Instance level.

10.7.3 Reporting Services Reports

If a Reporting Services database exists on a SQL Server, the Reporting Services node will be displayed under the SQL Server Agent node, under which all reports are listed by name. Expand the node to view the list of reports, or single-click the node to view a calendar for all reports. The Reporting Services event provider is used behind the scenes to access report metadata and history information.

For scheduled reports, Reporting Services jobs are listed under the Jobs node using a combination of the report name and schedule type. Keep in mind that not all reports have schedules, and some reports have multiple schedules, so the associated object nodes shown under Jobs and Reporting Services nodes may not match exactly. "Shared" report schedules will be listed under the Jobs node.

Reports are displayed on the Calendar by a friendly name, and offer many details through their popup window.
10.7.4 SQL Server Agent Log

The SQL Server Agent Log node is listed under each SQL Server Instance node. Single-click the node to view a calendar for all SQL Server Agent events. The SQL Server Agent event provider is used behind the scenes to access this history information.

10.7.5 Windows Tasks

**Win Sentry** supports the Task Scheduler Event Source. Windows Tasks are listed for each Windows Instance under the Tasks node. Expand the node to view the list of tasks, or single-click the node to view a calendar for all tasks. The Windows Task Scheduler event provider is used behind the scenes to access task metadata, history, and active status information.

**IMPORTANT:** Windows Vista introduced Task Scheduler 2.0. Task Scheduler 2.0 is backwards compatible with Task Scheduler 1.0, however, Task Scheduler 1.0 is not forwards compatible with Task Scheduler 2.0. For this reason, in order to Watch or Synchronize Task Scheduler 2.0 connections, you must have a SentryOne Monitoring Service and SentryOne Client running Windows Vista or higher.

Windows 8 and Windows 2012 also introduced changes to Task Scheduler. In order to Watch or Synchronize Windows 8 and Windows Server 2012 Instances, you must have a SentryOne Monitoring Service and SentryOne Client running Windows 8 or Windows 2012.
SPECIFYING NON-ZERO SUCCESS CODES FOR WINDOWS TASKS

Some Task Scheduler tasks do not return "0" for success. In these cases, Win Sentry may misinterpret this non-zero exit code as a task failure. In order to prevent these "false negatives", you may specify an exact text string that Win Sentry is to recognize as the success code for that particular task. To do so, click on the task in the Navigator pane and enter the value next to "Exit code if task was successful".

NOTE: Occasionally, Windows Tasks may show as running even after they have completed. For this reason, while a task is running, the Close Running Event context menu item is available to manually close this event. This menu item should only be used when you are sure the task has completed, but shows as running in Win Sentry.

10.7.6 Windows Event Logs

Win Sentry includes the Windows Event Logs Event Source. The monitoring of the Application, Security, and System Event logs is supported. When you monitor a Windows Instance the Application and System Logs are initially watched. If you would like to monitor the Security log, simply select Watch Windows Event Log from the Security node’s context menu.

The Windows Event Logs Source has a default History Filter in place. The default filter is tailored to capture information about both SentryOne and SQL Server. With the History Filter you define rules which control exactly what types of events the Monitoring Service writes into Event History concerning the Windows Event logs. If an event does not meet the criteria you have defined in the History Filter, information about that event is not written to your SentryOne Database. You may change the filter as needed to best fit your environment.

The Windows Event Logs History Filter can be accessed in the Settings pane as follows:

1. Select the Windows Instance node in the Navigator pane.
2. Open the Settings pane (View menu -> Settings).
3. Select Windows Event Logs Source from the drop down menu.

Note: Monitoring the Windows Event Logs is supported for Windows Vista or higher.

10.8 Event Objects

At a high level, an event object is simply any job, task, or other event type that can be managed using SentryOne. There are two general types of events: active event objects and status events. The two are actually completely different, with the exception that their event instances both occur at some measurable point in time, and therefore can be represented chronologically in calendar and list views. A third sub-type is hybrid event objects which contain some characteristics of both active and/or status events.

WHY IS IT IMPORTANT TO DIFFERENTIATE THE TYPES OF EVENT
When you view a calendar in SentryOne, you will see a combination of various types of active and status event instances. Active event objects include SQL Agent Jobs, Maintenance Plans, and Reporting Services Reports, along with Windows Task Scheduler Tasks. Status events include SQL Agent Alerts and SQL Agent Log records. Since busy calendars can include some or all of these event types, it’s important to be able to differentiate them to make the most of the information presented. The SentryOne Client facilitates this by indicating the type of event object using a small **glyph** on each event instance, as well as in event instance popup windows.

It’s also important to understand why, for example, active event objects have runtime and performance graphs, as well as runtime and performance conditions, while status events do not.

<table>
<thead>
<tr>
<th>Event Objects</th>
<th>Type</th>
<th>Instance Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Agent Job</td>
<td><strong>Active</strong></td>
<td>SQL Server</td>
</tr>
<tr>
<td>SQL Server Agent Alert</td>
<td><strong>Status (hybrid)</strong></td>
<td>SQL Server</td>
</tr>
<tr>
<td>SQL Server Agent Log Record</td>
<td><strong>Status</strong></td>
<td>SQL Server</td>
</tr>
<tr>
<td>Maintenance Plan</td>
<td><strong>Active (hybrid)</strong></td>
<td>SQL Server</td>
</tr>
<tr>
<td>Reporting Services Report</td>
<td><strong>Active</strong></td>
<td>SQL Server</td>
</tr>
<tr>
<td>Windows Task</td>
<td><strong>Active</strong></td>
<td>Windows</td>
</tr>
<tr>
<td>Windows Event Logs Event</td>
<td><strong>Status</strong></td>
<td>Windows</td>
</tr>
</tbody>
</table>

### 10.8.1 Active Event Objects

An **active event object** is also commonly referred to as a job or a task, and contains all of the information necessary to perform some measurable work. This can include instructions in some scripting language (T-SQL, VBScript, etc.), connection information, configuration parameters, etc. Active events always have a start time and an end time, i.e., they have duration, and they are also schedulable. SQL Agent Jobs and Windows Tasks are examples active event objects.

### 10.8.2 Status Events

A **status event** represents the discrete occurrence of some condition of a process, usually a persistent process. Status events do not have duration, they are not schedulable, and they do not perform any measurable work. For example, the SQL Server Agent Log contains various status events given off by the SQL Server process at specific points in time.

**STATUS EVENT OBJECTS**

A **status event object** is any status event that is predefined and/or configurable, such as a SQL
Server Agent error condition. Most entries in the SQL Agent Logs are instances of a status event object, such as this one:

[264] An attempt was made to send an email when no email session has been established

Although this event object is not “configurable”, it represents an error condition that is predefined and static.

**AD HOC STATUS EVENTS**

Not all status events can be pre-defined, for example an “ad hoc” message for a transaction log backup recorded in the SQL Server Log:

Log backed up: Database: Northwind, creation date(time): 2004/07/27(11:51:24), first LSN: 238:292:1, last LSN: 238:292:1, number of dump devices: 1, device information: (FILE=1, TYPE=DISK: {'D:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\

In this case a discrete status event is logged as the result of some process, but does not specify a start or end time, nor is the output constant, i.e., it changes every time. It is also not entirely pre-defined nor is it configurable; therefore we consider this an ad hoc status event.

### 10.8.3 Hybrid Event Objects

**Hybrid event objects** are those that don’t necessarily fall cleanly into the active or status event categories.

For example, a SQL Agent Maintenance Plan is an active event object that is comprised of several other active event objects – SQL Agent jobs. The plan itself doesn't actually do any work, the jobs do; they handle the various activities associated with the plan such as backups, optimizations, etc. In this case the jobs act as surrogates for the plan itself. The plan does, however, contain configuration information, as well as an execution engine and log, and its associated jobs do work and have duration. Therefore we consider the plan a variant of an active event object. If it did not have its own configuration info or execution log, it would simply be a collection of jobs.

A SQL Agent Alert is a variant of a status event object. Alerts can be triggered by various conditions of the SQL Server process, such as a full transaction log, a corrupt database, etc. SQL Server logs a status event to the Windows Application Log, then SQL Agent picks it up and fires any defined Alerts that match the condition. Although Alerts don't actually have their own log, when they fire SentryOne is able to capture the output as a discrete status event instance and log it to a table in MSDB, effectively an event log. SQL Agent Alerts are really more of a tool for the DBA to define the events for which they want to be notified or take some other action -- they aren't really status event objects in the strictest sense, but for our purposes it helps to think of them that way.

### 10.8.4 Event Instances

An event instance is any individual historical or future occurrence of an active event object or status event. Event instances correlate to the rectangular placeholders displayed on a SentryOne calendar, or the master records shown in a list view. An individual execution of a SQL Agent job
represents an active event instance. Every occurrence of a SQL Agent error in the SQL Agent logs is an individual status event instance.
When a SQL Server Instance is first set to watched status with SQL Sentry, SentryOne auto-installs a stored procedure in the MSDB database named **sp_sentry_mail_20** that is intended to be an interface compatible replacement for xp_sendmail. **sp_sentry_mail_20** supports almost all of the functionality of xp_sendmail, including sending query results as attachments.

Aside from using SMTP, not MAPI, **sp_sentry_mail_20** is a standard stored procedure, not an extended stored procedure, so it doesn't require installation of any dll's, scripts, or other components on every SQL Server. Thus, the configuration and maintenance requirements are dramatically reduced over xp_sendmail and other xp_sendmail replacements. In addition, because **sp_sentry_mail_20** has no dependencies on MAPI or SQLMail it is inherently more reliable than xp_sendmail.

After watching the SQL Server Instance, all that's required is replacing any calls to xp_sendmail with **sp_sentry_mail_20** as shown below.

**NOTE:** For **sp_sentry_mail_20** to function, **SMTP settings** must be configured first.

### SP_SENTRY_MAIL_20 PARAMETERS

A list of **sp_sentry_mail_20** parameters is shown in the table below. All parameters are optional except for @recipients. Parameters are identical to those used by xp_sendmail, with the exception of two new optional parameters (@from_address and @email_format).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@recipients</td>
<td>Email addresses of recipients, delimited by semicolons.</td>
</tr>
<tr>
<td>@message</td>
<td>The body of the message.</td>
</tr>
<tr>
<td>@query</td>
<td>A SQL statement to be executed. The results of the query will be included</td>
</tr>
<tr>
<td></td>
<td>in the body of the email by default, or as an attachment if @attach_results</td>
</tr>
<tr>
<td>@attachments</td>
<td>A semi-colon delimited list of files to attach to the message. <strong>Currently ignored. Planned for future version.</strong></td>
</tr>
<tr>
<td>@copy_recipients</td>
<td>Email addresses of recipients to be copied, delimited by semicolons.</td>
</tr>
<tr>
<td>@blind_copy_recipients</td>
<td>Email addresses of recipients to be blind copied, delimited by semicolons.</td>
</tr>
<tr>
<td>@subject</td>
<td>Subject line of the message.</td>
</tr>
<tr>
<td>@type</td>
<td>Ignored. used by xp_sendmail only, but provided for xp_sendmail compatibility.</td>
</tr>
<tr>
<td>@attach_results</td>
<td>If TRUE, the query results will be attached as a text file.</td>
</tr>
<tr>
<td></td>
<td>If FALSE (default), the results will be included inline in the body of the</td>
</tr>
<tr>
<td></td>
<td>message.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>@no_output</td>
<td>If TRUE, no output is returned to the client when the message is sent.</td>
</tr>
<tr>
<td></td>
<td>If FALSE (default), &quot;Mail sent.&quot; is returned to the client</td>
</tr>
<tr>
<td>@no_header</td>
<td>If FALSE (default), no column headers are included in the query results.</td>
</tr>
<tr>
<td>@width</td>
<td>The line width for rows in the query output. 0 (default) indicates no</td>
</tr>
<tr>
<td></td>
<td>width limit should be used.</td>
</tr>
<tr>
<td>@separator</td>
<td>Column separator value to use for the query output.</td>
</tr>
<tr>
<td>@echo_error</td>
<td>Ignored. used by xp_sendmail only, but provided for xp_sendmail compatibility.</td>
</tr>
<tr>
<td>@set_user</td>
<td>The user context under which to execute the query. Currently ignored.</td>
</tr>
<tr>
<td></td>
<td>Planned for future version.</td>
</tr>
<tr>
<td>@dbuse</td>
<td>The name of database in which to run the query.</td>
</tr>
<tr>
<td>@from_address</td>
<td>The &quot;mail from&quot; address to use for the message.</td>
</tr>
<tr>
<td>@email_format</td>
<td>Allowed values are 'text' or 'html' only.</td>
</tr>
</tbody>
</table>

**REPLACING CALLS TO XP_SENDMAIL**

Simply change calls to xp_sendmail to msdb..sp_sentry_mail_20 as shown below:

**Old call:**

```sql
eexec xp_sendmail @recipients = 'user@SentryOne.com', ...
```

**New call:**

```sql
eexec msdb..sp_sentry_mail_20 @recipients = 'user@SentryOne.com', ...
```

**EXAMPLE: RESULTS EMBEDDED IN MESSAGE BODY**

```sql
eexec msdb..sp_sentry_mail_20
    @recipients = 'user@SentryOne.com',
    @subject = 'Titles List',
    @Message = 'Titles List Report',
    @query = 'SELECT title_id, title FROM titles',
    @dbuse = 'pubs'
```

Results:
EXAMPLE: COMMA-SEPARATED RESULTS ATTACHED TO MESSAGE

exec msdb..sp_sentry_mail_20
    @recipients = 'user@SentryOne.com',
    @subject = 'Titles List',
    @Message = 'Titles List Report',
    @query = 'SELECT * FROM titles',
    @dbuse = 'pubs',
    @attach_results = 'TRUE',
    @separator = ',';

Results:

<table>
<thead>
<tr>
<th>title_id</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PC1005 But Is It User Friendly?</td>
</tr>
<tr>
<td>2</td>
<td>PS2067 Computer Phobic AND Non-Phobic Individuals: Behavior Variations</td>
</tr>
<tr>
<td>3</td>
<td>BU1111 Cooking with Computers: Surreptitious Balance Sheets</td>
</tr>
<tr>
<td>4</td>
<td>PS7777 Emotional Security: A New Algorithm</td>
</tr>
<tr>
<td>5</td>
<td>TC4203 Fifty Years in Buckingham Palace Kitchens</td>
</tr>
<tr>
<td>6</td>
<td>PS2067 Is Anger the Enemy?</td>
</tr>
<tr>
<td>7</td>
<td>PS2106 Life Without Fear</td>
</tr>
<tr>
<td>8</td>
<td>PCS009 Net Etiquette</td>
</tr>
<tr>
<td>9</td>
<td>TC3210 Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean</td>
</tr>
<tr>
<td>10</td>
<td>PS3333 Prolonged Data Deprivation: Four Case Studies</td>
</tr>
<tr>
<td>11</td>
<td>PC6606 Secrets of Silicon Valley</td>
</tr>
<tr>
<td>12</td>
<td>MC2222 Silicon Valley Gastronomic Treats</td>
</tr>
<tr>
<td>13</td>
<td>BUTF52 Straight Talk About Computers</td>
</tr>
<tr>
<td>14</td>
<td>TC7777 Sushi, Anyone?</td>
</tr>
<tr>
<td>15</td>
<td>BUTF52 The Busy Executive's Database Guide</td>
</tr>
<tr>
<td>16</td>
<td>MC3021 The Gourmet Microwave</td>
</tr>
<tr>
<td>17</td>
<td>MC9026 The Psychology of Computer Cooking</td>
</tr>
<tr>
<td>18</td>
<td>BUTF52 You Can Combat Computer Stress!</td>
</tr>
</tbody>
</table>
When a SQL Server 2005 and above Instance is first set to watched status with SQL Sentry, SentryOne auto-installs a stored procedure in the MSDB database named `sp_sentry_dbmail_20` that is intended to be an interface compatible replacement for `sp_send_dbmail`. `sp_sentry_dbmail_20` supports all of the functionality of `sp_send_dbmail`, without the need to configure each server to use it. Thus, the configuration and maintenance requirements are dramatically reduced over `sp_send_dbmail`.

After watching the SQL Server 2005 and above Instance, all that’s required is replacing any calls to `sp_send_dbmail` with `sp_sentry_dbmail_20` as shown below.

**NOTE:** For `sp_sentry_dbmail_20` to function **SMTP settings** must be configured first.

### `SP_SENTRY_DBMAIL_20` PARAMETERS

A list of `sp_sentry_dbmail_20` parameters is shown in the table below. All parameters are optional except for `@recipients`. You must specify at least one of `@body`, `@query`, `@file_attachments`, or `@subject`. Otherwise, `sp_sentry_dbmail_20` returns an error. Parameters are identical to those used by `sp_send_dbmail`.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@profile_name</td>
<td>Profiles are required to be set up in SQL Server 2005 and above before using mail functionality. Profiles are NOT required for SentryOne, therefore this parameter is optional for <code>sp_sentry_dbmail_20</code>.</td>
</tr>
<tr>
<td>@recipients</td>
<td>Email addresses of recipients, delimited by semicolons.</td>
</tr>
<tr>
<td>@copy_recipients</td>
<td>Email addresses of recipients to be copied, delimited by semicolons.</td>
</tr>
<tr>
<td>@blind_copy_recipients</td>
<td>Email addresses of recipients to be blind copied, delimited by</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>@subject</td>
<td>Subject line of the message.</td>
</tr>
<tr>
<td>@body</td>
<td>The body of the message.</td>
</tr>
<tr>
<td>@body_format</td>
<td>The format of the message body. May be TEXT or HTML.</td>
</tr>
<tr>
<td>@importance</td>
<td>The importance of the message. May be Low, Normal, or High.</td>
</tr>
<tr>
<td>@sensitivity</td>
<td>The sensitivity of the message. May be Normal, Personal, Private, or Confidential.</td>
</tr>
<tr>
<td>@file_attachments</td>
<td>A semi-colon delimited list of files to attach to the message.</td>
</tr>
<tr>
<td>@query</td>
<td>A SQL statement to be executed. The results of the query will be included in the body of the email by default, or as an attachment if @attach_query_result_as_file = 1.</td>
</tr>
<tr>
<td>@execute_query_database</td>
<td>The name of database on which to run the query.</td>
</tr>
<tr>
<td>@attach_query_result_as_file</td>
<td>If 1, the query results will be attached as a text file. If 0 (default), the results will be included inline in the body of the message.</td>
</tr>
<tr>
<td>@query_attachment_filename</td>
<td>The file name to use for the result set of a query when attached. If none is specified and the above parameter =1, an arbitrary file name will be used.</td>
</tr>
<tr>
<td>@query_result_header</td>
<td>If 1(default), the query results will include column headers. If 0, they will not.</td>
</tr>
<tr>
<td>@query_result_width</td>
<td>The line width, in characters, for rows in the query output.</td>
</tr>
<tr>
<td>@query_result_separator</td>
<td>Column separator value to use for the query output. The default is ' ' (space)</td>
</tr>
<tr>
<td>@exclude_query_output</td>
<td>If 1, no output is returned to the client when the message is sent. If 0 (default), &quot;Mail sent.&quot; is returned to the client</td>
</tr>
<tr>
<td>@append_query_error</td>
<td>If 1, the message is sent with the query error in the body. If 0 (default), the message is not sent.</td>
</tr>
<tr>
<td>@query_no_truncate</td>
<td>If 1, query results are not truncated. If 0 (default), columns truncate to 256 characters.</td>
</tr>
<tr>
<td>@mailitem_id</td>
<td>Optional output parameter returns the mailitem_id of the message. The mailitem_id is of type int.</td>
</tr>
<tr>
<td>@from_address</td>
<td>The &quot;mail from&quot; address to use for the message.</td>
</tr>
</tbody>
</table>
REPLACING CALLS TO SP_SEND_DBMAIL

Simply change calls to sp_send_dbmail to msdb..sp_sentry_dbmail_20 as shown below:

Old call:

   exec sp_send_dbmail @recipients = 'user@intercerve.com', ...

New call:

   exec msdb..sp_sentry_dbmail_20 @recipients = 'user@intercerve.com', ...

EXAMPLE: RESULTS EMBEDDED IN MESSAGE BODY

exec msdb..sp_sentry_dbmail_20
   @recipients = 'user@intercerve.com',
   @subject = 'Titles List',
   @body = 'Titles List Report',
   @query = 'SELECT title_id, title FROM titles',
   @execute_query_database = 'pubs'

Results:
EXAMPLE: COMMA-SEPARATED RESULTS ATTACHED TO MESSAGE

exec msdb..sp_sentry_dbmail_20
  @recipients = 'user@intercervce.com',
  @subject = 'Titles List',
  @body = 'Titles List Report',
  @query = 'SELECT * FROM titles',
  @execute_query_database = 'pubs',
  @attach_query_result_as_file = 1,
  @query_result_separator = ',',
  @query_attachment_filename = 'queryresults2.txt'

Results:

Attachments: 1 queryresults2.txt (2.1K)

Titles List Report

<table>
<thead>
<tr>
<th>title_id</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC1035</td>
<td>But Is It User Friendly?</td>
</tr>
<tr>
<td>PS1372</td>
<td>Computer Phobic AND Non-Phobic Individuals: Behavior Variations</td>
</tr>
<tr>
<td>BU1111</td>
<td>Cooking with Computers: Surreptitious Balance Sheets</td>
</tr>
<tr>
<td>PS1777</td>
<td>Emotional Security: A New Algorithm</td>
</tr>
<tr>
<td>TC4203</td>
<td>Fifty Years in Buckingham Palace Kitchens</td>
</tr>
<tr>
<td>PS2031</td>
<td>Is Anger the Enemy?</td>
</tr>
<tr>
<td>PS2106</td>
<td>Life Without Fear</td>
</tr>
<tr>
<td>PC9999</td>
<td>Net Etiquette</td>
</tr>
<tr>
<td>TC3218</td>
<td>Onions, Leeks, and Garlic: Cocking Secrets of the Mediterranean</td>
</tr>
<tr>
<td>PS3333</td>
<td>Prolonged Data Deprivation: Four Case Studies</td>
</tr>
<tr>
<td>PC9988</td>
<td>Secrets of Silicon Valley</td>
</tr>
<tr>
<td>MC3322</td>
<td>Silicon Valley Gastronomic Treats</td>
</tr>
<tr>
<td>BU7332</td>
<td>Straight Talk About Computers</td>
</tr>
<tr>
<td>TC7777</td>
<td>Sushi, Anyone?</td>
</tr>
<tr>
<td>BU1032</td>
<td>The Busy Executive's Database Guide</td>
</tr>
<tr>
<td>MC3321</td>
<td>The Gourmet Microwave</td>
</tr>
<tr>
<td>MC3326</td>
<td>The Psychology of Computer Cooking</td>
</tr>
<tr>
<td>BU3375</td>
<td>You Can Combat Computer Stress!</td>
</tr>
</tbody>
</table>
11.3 Watched Server Objects

**SENTRYONE WATCHED SERVER OBJECTS**

When a SQL Server instance is watched by SentryOne, the below objects are placed on the target server. To remove these objects, see the [Uninstalling SentryOne topic](#).

**TABLES (MSDB):**

- SQLSentryObjectVersion_20
- SQLSentryAlertLog_20
- SQLSentryDBEmails_20 (SQL Server 2005+)
- SQLSentryDBEmail_Attachments_20 (SQL Server 2005+)
- SQLSentryEmails_20
- SQLSentryLogCache_20
- SQLSentryLogData_20
- SQLSentryObjectVersion_20
- SQLSentryQueueLog_20

**STORED PROCEDURES (MSDB):**

- spGetBlockInfo_20
- spGetProcedureStatsData
- spGetQueryStatsData
- spGetBlockInfo_20
spGetJobInfo_20
spQueueHeartbeat_20
spQueueJob_End_20
spQueueJob_Start_20
spQueueMonitor_20
spReadLogFile_20
spSetupAlertsTrap_20
spTrapAlert_20
sp_sentry_mail
sp_sentry_mail_20
sp_sentry_dbmail_20 (SQL Server 2005+)

**SQL AGENT JOBS:**
- SQLSentry 2.0 Alert Trap
- SQLSentry 2.0 Queue Monitor

**SENTRYONE MONITORING OBJECTS IN AZURE SQL DATABASE TARGETS**

When an Azure SQL Database is watched by DB Sentry, the user is given the option to allow some monitoring objects to be created in the target database. If these objects are allowed they will be created under a SentryOne schema. To remove these objects, see the Uninstalling SentryOne topic. Some of the tables will have unique identifiers on them after the table name (indicated with'...').

**TABLES (TARGET DATABASE)**
- SQLSentry.SQLSentryObjectVersion_20
- SQLSentry.ProcedureStats...
- SQLSentry.ProcedureStats...

**STORED PROCEDURES (TARGET DATABASE)**
- SQLSentry.spGetProcedureStatsData
- SQLSentry.spGetQueryStatsData

11.4 **Execute Action Parameters**

Execute SQL and Execute Process actions allow for the passing of parameters in the command text.

**SYSTEM PARAMETERS**

There are nine system parameters that are available in SentryOne. Below is a list of the system parameters along with a description of the information they hold.

<table>
<thead>
<tr>
<th>System Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ServerName</strong>*</td>
<td>Name of the Instance on which the object exists</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><strong>ObjectName</strong></td>
<td>Name of the object</td>
</tr>
<tr>
<td><strong>ObjectType</strong></td>
<td>SQL Server Agent Job, Maintenance Plan, etc.</td>
</tr>
<tr>
<td><strong>Owner</strong>*</td>
<td>Owner of the object</td>
</tr>
<tr>
<td><strong>StepName</strong>*</td>
<td>Name assigned to the step</td>
</tr>
<tr>
<td><strong>Message</strong>*</td>
<td>Output text associated with the step</td>
</tr>
<tr>
<td><strong>Category</strong>*</td>
<td>Category as defined in the properties of the object</td>
</tr>
<tr>
<td><strong>Condition</strong></td>
<td>Failure, Completed, etc.</td>
</tr>
<tr>
<td><strong>MessageText</strong></td>
<td>Providing the body text of the standard SentryOne notification.</td>
</tr>
</tbody>
</table>

* Available for General Actions only.

These parameters are passed to the command text by surrounding the parameter name with `<% %>`, such as `<%ServerName%>`.

**USER PARAMETERS**

User defined parameters can also be defined in the output text of an event. The syntax within the output must be: PARAM=12345, PARAM: 12345, or PARAM:12345. You would then use `<%PARAM%>` within the command text of the execute action to pass that value.

**EXAMPLE**

Your organization keeps a table on its Issues database called FailedAccountingJobs, which has columns for JobOwner, JobName, LastOrderProcessed, and Time. The Accounting jobs create step output that records the last OrderID processed in case of failure using OrderID as the user defined parameter. You could set up an Execute SQL action on the SQL Agent Job: Failure condition that points to the server that holds the Issues database with a T-SQL command:

```sql
Insert Into Issues..FailedJobs (JobOwner, JobName, LastOrderProcessed, Time)
VALUES ('%Owner%', '%ObjectName%','%OrderID%', GetDate())
```

The values for the owner, name of the object, and OrderID will automatically be provided by SentryOne and inserted into the command when it runs.

**NOTE:**

- User defined parameters may not be retrievable from the output of retried SQL Agent Job steps.
When passing the MessageText parameter, double quotes may be needed. Also remember to use SET QUOTED_IDENTIFIER OFF in this case.

**ADDITIONAL OPTIONS**

Other options are available for the StepName and Message parameters. The StepName and Message can also be passed using the format `<%StepName:[Identifier]%>` or `<%Message:[Identifier]%>`. If the [Identifier] is a number then the value from that particular step ID is used, starting with step 1.

Additionally, the [Identifier] can also be one of the following values:

- FirstStep – The value for the step name or message of the first completed step.
- LastStep – The value for the step name or message of the last completed step.
- FirstFailure – The value for the step name or message of the first failed step.
- LastFailure – The value for the step name or message of the last failed step.

**Example:** INSERT INTO TABLE (MyCustomMessage) VALUES ('The last step failure was on step `<%StepName:LastFailure%>` and the text for that was `<%Message:LastFailure>'

11.5 SentryOne Database Maintenance

Just as with any other SQL Server database, it is important that regular maintenance activities be performed on the SentryOne database to ensure optimal performance. The following are suggested practices for performing such maintenance.

**WEEKLY**

**INDEX RE BuildS**

Weekly Index rebuilds are recommended. SentryOne Fragmentation Manager can be used to help you make intelligent decisions about index management in your environment. For more information see the Fragmentation Manager topic.

**NIGHTLY**

**DATABASE INTEGRITY CHECKS**

Database integrity checks should be performed every night. They also can be scheduled through a maintenance plan and will then be reflected in the SentryOne Client.

**INDEX D EFRAGMENTSATION**
**SentryOne Fragmentation Manager** can be used to help you make intelligent decisions about index management in your environment. For more information see the Fragmentation Manager topic.

**CONFIGURING THE SENTRYONE DATABASE MAINTENANCE WINDOW**

The Maintenance Window should also be configured with the above activity scheduled within to avoid excessive Failsafe Notifications due to these activities.

### 11.6 Relocating the SentryOne Database

Moving the **SentryOne Database** to a new SQL Server instance is a straightforward process. Following are the steps required to accomplish this:

1. Retrieve the instance name for the new SQL Server by running `SELECT SERVERPROPERTY('SERVERNAME')` against it. Your license key is based on this instance name.
   a. In the Perpetual Licenses section you will see your SentryOne License listed. **Click the plus symbol** next to the instance name to expand the license management options.
   b. Select the License Configuration tab.
   c. In the SQL Server Instance Name field enter the new instance name from Step 1.
   d. In the Reason For Change field enter a simple reason for the change.
   e. **Click the Update button** and the license will be automatically emailed to you.
3. Once you have received the new key, shut down all SentryOne Clients and SentryOne Monitoring Services.
4. Perform a full backup of the SentryOne Database.
5. Detach the SentryOne Database.
6. Move the associated data and log files.
7. Attach the SentryOne Database to the new server.
8. Open the SentryOne Client on any machine with a SentryOne Monitoring Service. After failing to locate the SentryOne Database, the **Connect** window should appear. Enter the new location of the SentryOne Database. The SentryOne Client will alert you that it needs to be restarted to connect to the new SentryOne Database. On restarting, you will be asked for the new license key.
9. After successfully applying the license and starting the SentryOne Client, you will need to run the Service Configuration Utility, pointing the Monitoring Service at the new SentryOne Database. The Service Configuration Utility can be found in the SentryOne program group. After running the Service Configuration Utility be sure that the Monitoring Service has been restarted.
10. Repeat steps 8 and 9 on any other machines with SentryOne Monitoring Services. You will not be asked for the license again, however.

You may now start any other SentryOne Client and enter the new SentryOne Database location when asked.

Don’t forget to setup **SentryOne Database maintenance** on the new server.
To remove an old server from the All Instances View so that you will no longer be prompted about it when opening the SentryOne Client:

1. Select the Server Name of the server you would like to remove in the “Please Specify the Server Name” drop-down list, and press Shift + Delete.
2. The new server should now be the only one listed in the drop-down list. Use the Connect command and connect to the new installation.

The next time you open the Client it will automatically connect to the new installation.

11.7 Watching Servers Across Domains

It is possible to watch Instances across domains with SentryOne even when there is no trust relationship between them. The best option to achieve this depends on the resources available and number of servers you wish to watch.

**ONE MONITORING SERVICE**

If you only have the resources to install one SentryOne Monitoring Service for your environment, or only have a couple servers in non-trusted domains you wish to watch, pass-through authentication can be set up on each server in the other domain. This requires each watched server on the other domain to have a local Windows account that has the identical login and password as the SentryOne Monitoring Service service's domain account. This must also have sysadmin privileges on the watched SQL Server instance.

**MULTIPLE SERVER SERVICES**

Another option is to install a SentryOne Monitoring Service in each domain where you have servers you wish to watch. This only requires pass-through authentication to be set up for each Monitoring Service to the machine where the SentryOne Database is installed. You could then create separate sites for each Monitoring Service to ensure that they only polled the servers in their domain.

11.8 Hot Key List

Below are the hot keys that are available in SentryOne:

<table>
<thead>
<tr>
<th>Item</th>
<th>Keys</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td>F1 or Ctrl + H</td>
<td>Opens the User Guide.</td>
</tr>
<tr>
<td>Save</td>
<td>Ctrl + S</td>
<td>Saves changes made in the client.</td>
</tr>
<tr>
<td>Close</td>
<td>Ctrl + W</td>
<td>Closes the active tab in the client.</td>
</tr>
<tr>
<td>Exit</td>
<td>Ctrl + Q</td>
<td>Closes the client.</td>
</tr>
<tr>
<td>Item</td>
<td>Keys</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td><strong>User Preferences</strong></td>
<td>Ctrl + O</td>
<td>Opens the User Preferences window.</td>
</tr>
<tr>
<td><strong>Print</strong></td>
<td>Ctrl + P</td>
<td>Opens the Print window.</td>
</tr>
<tr>
<td><strong>Find Object</strong></td>
<td>Ctrl + F</td>
<td>Opens the Event Object Search window.</td>
</tr>
<tr>
<td><strong>Tab Scroll</strong></td>
<td>Ctrl + Tab</td>
<td>Scrolls through open tabs in the client.</td>
</tr>
<tr>
<td><strong>Select All</strong></td>
<td>Ctrl + A</td>
<td>Selects all items in window (where applicable).</td>
</tr>
<tr>
<td><strong>Copy</strong></td>
<td>Ctrl + C</td>
<td>Copies selected items to clipboard (where applicable).</td>
</tr>
<tr>
<td><strong>Paste</strong></td>
<td>Ctrl + V</td>
<td>Pastes items from clipboard (where applicable).</td>
</tr>
<tr>
<td><strong>Add Note</strong></td>
<td>Ctrl + Alt + N</td>
<td>Opens the notes interface when an applicable object is selected. On note dialog:  - New  - Ctrl + N  - Save  - Ctrl + S  - Delete  - Ctrl + D  - Close  - Ctrl + C</td>
</tr>
<tr>
<td><strong>History List (one day)</strong></td>
<td>Ctrl + Alt + H</td>
<td>Opens the history List view for one day when an applicable object is selected. On note dialog:  - Opens new note.  - Saves note.  - Deletes note.  - Closes note dialog.</td>
</tr>
</tbody>
</table>
11.9 SentryOne PowerShell Module

The SentryOne installation package includes a PowerShell module which can be used to manage your SentryOne environment through PowerShell. This topic includes a walkthrough of that functionality.

**NOTE:** PowerShell 3.0 or higher is required to use this functionality.

**GETTING STARTED**

The first step is to import the SentryOne PowerShell module. You can find the module in the SentryOne program directory. Use the **Import-Module** command.

```
Import-Module "C:\Program Files\SentryOne\11.0\Intercede.SQLSentry.PowerShell.psd1"
```

To verify that the module imported correctly, run the **Get-Module** command. You should see a listing for Intercerve.SentryOne.Powershell.

**COMMANDS**

Various commands are available, see the table below for more information about each command.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connect-SQLSentry</strong></td>
<td>This command allows you to connect to a specific SentryOne installation, which is required before any other actions can be performed. This command is also useful for navigating between repositories. Used in environments which have more than one SentryOne repository.</td>
</tr>
</tbody>
</table>

**Available Parameters:**

- **DatabaseName**
- **Login**
- **Password**
- **ServerName**
- **UseIntegratedSecurity**

**Disconnect-SQLSentry**

Available Parameters:
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-SQLSentryConfiguration</td>
<td>View Basic information about your SentryOne Configuration</td>
</tr>
<tr>
<td></td>
<td>View Connections in your environment</td>
</tr>
<tr>
<td>Available Parameters:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ConnectionType</td>
</tr>
<tr>
<td></td>
<td>ID</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>ObjectID</td>
</tr>
<tr>
<td>Get-Connection</td>
<td>View Computers in your environment</td>
</tr>
<tr>
<td>Available Parameters:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ComputerType</td>
</tr>
<tr>
<td></td>
<td>ID</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td>Get-Computer</td>
<td>View Sites in your environment</td>
</tr>
<tr>
<td>Available Parameters:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ID</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>In order to be watched with SentryOne, you must first register a Computer</td>
</tr>
<tr>
<td>Get-Site</td>
<td></td>
</tr>
<tr>
<td>Available Parameters:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ComputerType</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>AccessLevel</td>
</tr>
<tr>
<td></td>
<td>DatabaseName</td>
</tr>
<tr>
<td></td>
<td>Login</td>
</tr>
<tr>
<td></td>
<td>Password</td>
</tr>
<tr>
<td></td>
<td>Port</td>
</tr>
<tr>
<td></td>
<td>TargetSite</td>
</tr>
<tr>
<td></td>
<td>UseIntegratedSecurity</td>
</tr>
<tr>
<td>Register-Computer</td>
<td></td>
</tr>
</tbody>
</table>
NOTE: The -AllowAzureRemoteObjectInstallation parameter is ONLY valid for Azure SQL Database targets.

Watch a Computer with SentryOne.

**Invoke-WatchComputer**

Available Parameters:

- **Computer**
- **LicenseMode**

In order to be watched with SentryOne, you must first register a Connection.

**Register-Connection**

Available Parameters:

- **ConnectionType**
- **Name**
- **Login**
- **Password**
- **Port**
- **TargetSite**
- **UseIntegratedSecurity**

Watch a Connection with SentryOne.

**Invoke-WatchConnection**

Available Parameters:

- **Connection**

Stop watching a Connection with SentryOne.

**Invoke-UnwatchConnection**

Available Parameters:

- **Connection**

**EXAMPLES**

```powershell
<# Import the SentryOne PowerShell Module#>
Import-Module "C:\Program Files\SentryOne\11.0\Intercerve.SQLSentry.Powershell.psd1"
<# Connect to a specific SentryOne Installation#>
Connect-SQLSentry -ServerName server.domain.com -DatabaseName SentryOne
<# Get Information about your SentryOne Installation #>
Get-SQLSentryConfiguration
<# Get Information about the Sites in your SentryOne Installation - use parameters to find information for a specific site #>
Get-Site
```
<# Get Information about the Connections (Instances) in your SentryOne Installation -use parameters to find information for a specific connection "#>
Get-Connection

<# Get Information about the Computers (Targets) in your SentryOne Installation -use parameters to find information for a specific connection "#>
Get-Computer

<# Register Computers (Targets), so that they can be watched in your environment"#>
Register-Computer -ComputerType Windows -Name server.domain.com -AccessLevel Full

<# Register a Target that cannot utilize Windows Authentication (e.g., Azure SQL Database)"#>
Register-Computer -ComputerType AzureSqlDatabase -Name example.database.windows.net -DatabaseName dbName -Login username -Password password -AccessLevel Full -UseIntegratedSecurity 0

<# Register Connections (Instances), so that they can be watched in your environment"#>
Register-Connection -ConnectionType SqlServerAnalysisServices -Name server.domain.com

<# Watch Windows Computer (Target) with Performance Analysis and Event Calendar | -Pipe in the Computer"#>
Get-Computer -Name server.domain.com -NamedServerComputerType Windows | Invoke-WatchComputer

<# Watch Hyper-V Host (Target) with Performance Analysis and Event Calendar (core-based licensing) | -Pipe in the Computer"#>
Get-Computer -Name server.domain.com -NamedServerComputerType Windows | Invoke-WatchComputer -LicenseMode CoreBased

<# Watch SQLServer Connection (Instance) with Performance Analysis and Event Calendar | -Pipe in the Connection "#>
Get-Connection -Name server.domain.com -NamedServerConnectionType SqlServer | Invoke-WatchConnection

<# Unwatch Windows computer (Target)"#>
Get-Computer -Name server.domain.com -NamedServerComputerType Windows | Invoke-UnwatchComputer

<# Unwatch SSAS connection"#>
Get-Connection -Name server.domain.com -NamedServerConnectionType SqlServerAnalysisServices | Invoke-UnwatchConnection

<# Unwatch SQLServer connection"#>
Get-Connection -Name server.domain.com -NamedServerConnectionType SqlServer | Invoke-UnwatchConnection

<# Unregister Computers (Targets)"#>
Register-User -FirstName Test -LastName user -Email tuser@test.net -PagerAddress tuser@testPager.net -Description Tester -Login domain\username
Get-User -FirstName Test
Get-User -Name "Test User"
Disable-User -Name "Test User"
Enable-User -Name "Test User"
Unregister-User -Name "Test User"

<# Unregister Users from Group"#>
Register-Group -Name "Test Group" -Description "A Group" -Login Domain\TestGroup
Get-Group -Name "Test Group"
Disable-Group -Name "Test Group"
Enable-Group -Name "Test Group"
Unregister-Group -Name "Test Group"

<# User cmdlets"#>
Get-User -Name "Test User" | Add-GroupUser -GroupName "Test Group"
Get-User -Name "Test User" | Remove-GroupUser -GroupName "Test Group"
CUSTOMER PORTAL

Access to the SentryOne Customer Portal is available around the clock, allowing you to retrieve a backup license key, expand your enterprise by adding more licenses or even modify an existing license key in the case of hardware changes. In addition, the Customer Portal is where product updates and documentation can be found. The page http://www.SentryOne.com/portal can be used to activate and log into your account.

SUPPORT

If you have any technical questions, or for help with installation or configuration issues, please don’t hesitate to contact us:

  - Email: support@SentryOne.com
  - Phone: 704-895-6241
  - Toll Free: 855-775-7733

SUPPORT FORUM

Additional information about SentryOne can be found at the SentryOne support forum: http://support.sentryone.com

DEMOS

To sign up for one of our regularly scheduled public webinar demos please visit: http://www.SentryOne.com/company/news-events#webinars

SALES

If you have any pre-sales questions, or would like to place an order, please contact our sales team directly:

  - Email: sales@SentryOne.com
  - Phone: 704-895-6241

FEEDBACK

We always welcome your feedback on this guide and SentryOne in general. Please email any feedback, ideas, or feature requests to support@SentryOne.com
Index

Action Behaviors, 109-111
Actions, 103-109
Actions Log, 61-63
Active Event Objects, 262
Adding Notes, 71-73
Advisory Conditions, 96-102
Advisory Conditions, 96-102
Alerting and Response System, 77-78
All Targets, 36-38
AlwaysOn, 217-224
Appendix
  Execute Action Parameters, 273-275
  Hot Key List, 277-279
  Relocating the SentryOne Database, 276-277
  Reporting, 64-71
  SentryOne Database Maintenance, 275-276
  sp_sentry_dbmail_20, 268-272
  sp_sentry_mail_20, 265-268
  Watched Server Objects, 272-273
  Watching Servers Across Domains, 277
Audit Conditions, 94-96
Baselining, 183-187
Blocking SQL, 233-236
Calendar View, 140-144
Clearing Failed Instances, 163-164
Client Alerts, 25-26
Client Interface, 15-16
Client Security, 8-11
cloud.sentryone.com, 247-249
Concepts, 6
Condition Settings, 80-84
Conditions, 78-80
Conditions Pane, 19-21
Conflict Viewer, 164-165
Connect to Installation, 75-76
Contact Information, 283
Contact Management, 42-45
Copy Job/Alert Wizard, 164
Custom Event Views, 145
Custom Event Views
    Custom Event Views, 145
    Leveling Backups Across a SAN or NAS, 149-151
    Working With Custom Event Views, 145-148
Dashboard, 177-181
Dashboard Resolutions/Retention, 181-182
Data Grids, 246-247
Deadlocks, 236-240
Default Site, 38-39
Disk Activity, 196-200
Disk I/O, 196-200
Disk latency, 196-200
Disk Reports, 200-203
Disk Space, 200-203
Enabling SSIS Logging, 171-172
Event Chains, 151-160
Event Chains
    Event Chains, 151-160
Event Connector Architecture, 253-254
Event Instances, 263-264
Event Object Information Window, 165-166
Event Objects, 261-262
Event Providers, 257
Event Sources, 257-258
Event Store Connections, 256-257
Event Stores, 256
Event View Pane, 18-19
Execute Action Parameters, 273-275
Execute Job Action, 103-109
Execute Process Action, 103-109
Execute SQL Action, 103-109
Execution Agents, 254-255
Execution Engines, 255-256
Extended Job Step Logging (SQL Server 2005+), 171
Extended Job Step Logging / SSIS Logging, 170
Failsafe Conditions, 92-94
Favorite Targets, 35-36
Filter Editor, 84-85
Filter Process Flowchart, 148-149
Filtering Process, 148
Fragmentation Manager, 229-232
General Conditions, 85-92
Global Calendar View, 36-38
Global View, 36-38
History Filter, 138-139
History Synchronization Failed, 25-26
Hot Key List, 277-279
How to Configure Actions, 103-109
Hybrid Event Objects, 263
Important Concepts, 253
Important Concepts
  Active Event Objects, 262
  Event Connector Architecture, 253-254
  Event Instances, 263-264
  Event Objects, 261-262
  Event Providers, 257
  Event Sources, 257-258
  Event Store Connections, 256-257
  Event Stores, 256
  Execution Agents, 254-255
  Execution Engines, 255-256
  Hybrid Event Objects, 263
  Important Concepts, 253
  Reporting Services Reports, 259-260
  SQL Server Agent Alerts, 258-259
  SQL Server Agent Jobs, 258
  SQL Server Agent Log, 260
Status Events, 262-263
Windows Tasks, 260-261
Index Analysis, 214-216
Indexes tab, 225-229
Installation and Maintenance, 7
Instances, 41-42
Inventory View, 60-61
Job Queueing, 160-163
Job Queueing
   Job Queueing, 160-163
Leveling Backups Across a SAN or NAS, 149-151
License Management, 33-35
License Management
   License Management, 33-35
List View, 144-145
Log To Database Action, 103-109
Log To Disk Action, 103-109
Log To Event Log Action, 103-109
Logs
   Actions Log, 61-63
Menus, 26-29
Message Editing, 111-117
Monitor Additional Targets and Instances, 73-75
Monitoring a SQL Server Instance, 73-75
Monitoring a Windows Target, 73-75
Monitoring an Analysis Services Instance, 73-75
Monitoring Service, 51
Monitoring Service
   Inventory View, 60-61
   Monitoring Service, 51
      Monitoring Service Load Balancing and Fault Tolerance, 61
      Monitoring Service Logon Account, 64
      Monitoring Service Settings, 51-55
      Service Configuration Utility, 63-64
      Site Configuration, 58-60
Monitoring Service Failover, 61
Monitoring Service Load Balancing, 61
Monitoring Service Load Balancing and Fault Tolerance, 61
Monitoring Service Logon Account, 64
Monitoring Service Settings, 51-55
Multi-server Administration (MSX/TSX), 166-167
Navigator Nodes, 35
Navigator Pane, 16-18
Object Groups, 48-51
Output Content Match, 136-137
Performance Analysis Overview, 243-245
Performance Analysis Toolbar, 245-246
Performance Metrics, 187-194
Performance Monitoring
  Filter Process Flowchart, 148-149
  Filtering Process, 148
Plan Diagram Options, 214
Plan Explorer (integrated), 208-211
Plan Explorer Sessions, 211-212
Processes, 194-196
Purge History Older Than, 51-55
Query Plans, 224-225
Quick Traces, 240-243
Relocating the SentryOne Database, 276-277
Reporting, 64-71
Reporting Services Reports, 259-260
Response Ruleset Examples, 132-136
Response Rulesets, 129-132
Rights Based Security, 8-11
Role Based Security, 11-13
Role Based Security
  Role Based Security, 11-13
Run QuickTrace Action, 103-109
Runtime Stats, 167-168
Scan for Configuration Changes, 39-41
Schedules and Windows, 45-48
Security Overview, 8
Send Email Action, 103-109
Send Page Action, 103-109
Send SNMP Trap Action, 103-109
SentryOne Client, 14

SentryOne Client
Adding Notes, 71-73
Calendar View, 140-144
Clearing Failed Instances, 163-164
Conflict Viewer, 164-165
Connect to Installation, 75-76
Copy Job/Alert Wizard, 164
Enabling SSIS Logging, 171-172
Event Object Information Window, 165-166
Extended Job Step Logging (SQL Server 2005+), 171
Extended Job Step Logging / SSIS Logging, 170
List View, 144-145
Monitor Additional Targets and Instances, 73-75
Multi-server Administration (MSX/TSX), 166-167
Output Content Match, 136-137
Runtime Stats, 167-168
SentryOne Client, 14

SentryOne Database Maintenance, 275-276

SentryOne Event Calendar, 138

SentryOne Event Manager
SentryOne Event Calendar, 138

SentryOne Performance Analysis, 173-177
Baselining, 183-187
Blocking SQL, 233-236
cloud.sentryone.com, 247-249
Dashboard, 177-181
Data Grids, 246-247
Deadlocks, 236-240
Disk Activity, 196-200
Disk Space, 200-203
Performance Metrics, 187-194
Plan Explorer (integrated), 208-211