

Azure SQL DB Blocking SQL

Last Modified on 21 June 2022

Applies to: The **Blocking** tab is available for SQL Server, Azure SQL Database, and AWS RDS targets.

SQL Sentry Portal: This feature is available in [SQL Sentry Portal](#). To learn more about configuring your environment to use the on-premises, browser-based option with your existing [SQL Sentry database](#), see the [SQL Sentry Portal](#) article.

🔗 See the [SQL Sentry Portal Blocking](#) article for an example of the **Blocking** feature in a browser.

In this article:

- [Overview](#)
- [Blocking SQL Collection](#)
- [Blocking SQL Retention](#)
- [Blocking SQL Alerting](#)
- [Display/Controls](#)
- [Blocking SQL Grid View](#)
- [Blocking SQL Metrics](#)

Overview

The **Blocking SQL** tab displays all SQL Server blocks that occurred during the active date range that 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.

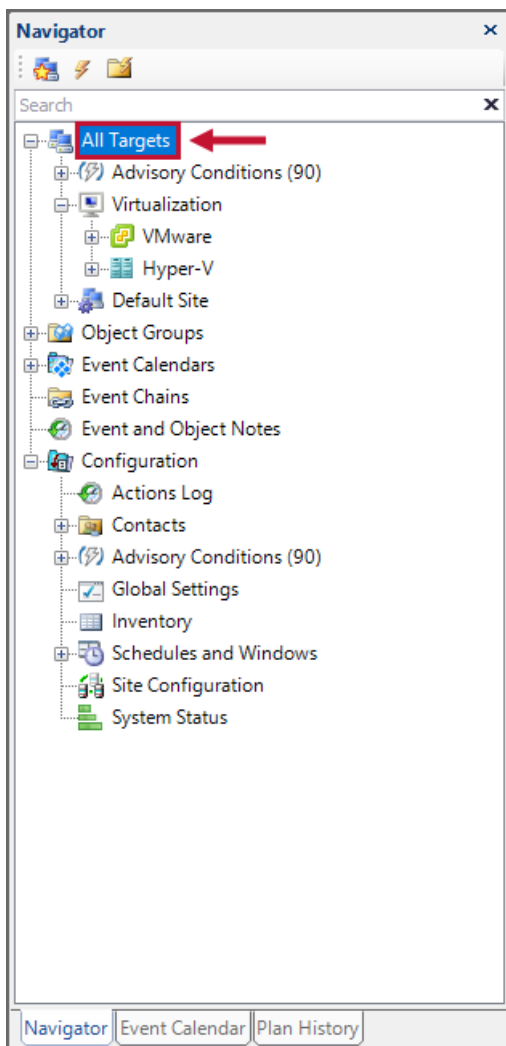
Blocking SQL Collection

Blocking SQL collection is controlled with the **Blocking SQL Source** settings. The **Minimum Block Duration** setting determines how long a block exists before information is collected on it. The default setting is 15 seconds. Adjust **Blocking Collection** as desired through the **Settings** pane (**View > Settings**).

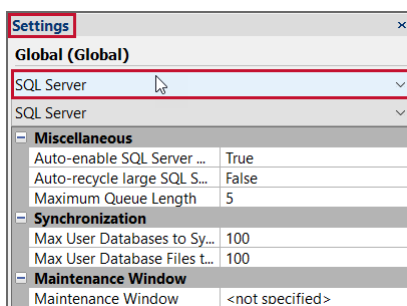
Example One

Change blocking collection to 20 seconds globally by completing the following steps:

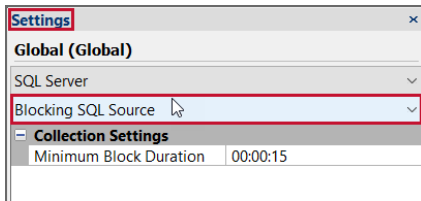
1. Select the **All Targets** (globally) node in the **Navigator** pane (**View > Navigator**).



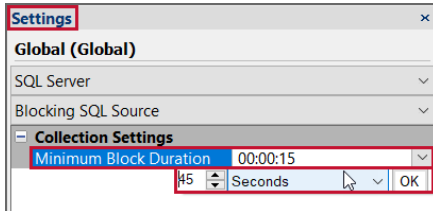
2. In the **Settings** pane (**View > Settings**), select **SQL Server** from the top drop-down list.



3. Select **Blocking SQL Source** from the second drop-down list to show **Blocking Source** settings that are being applied globally.



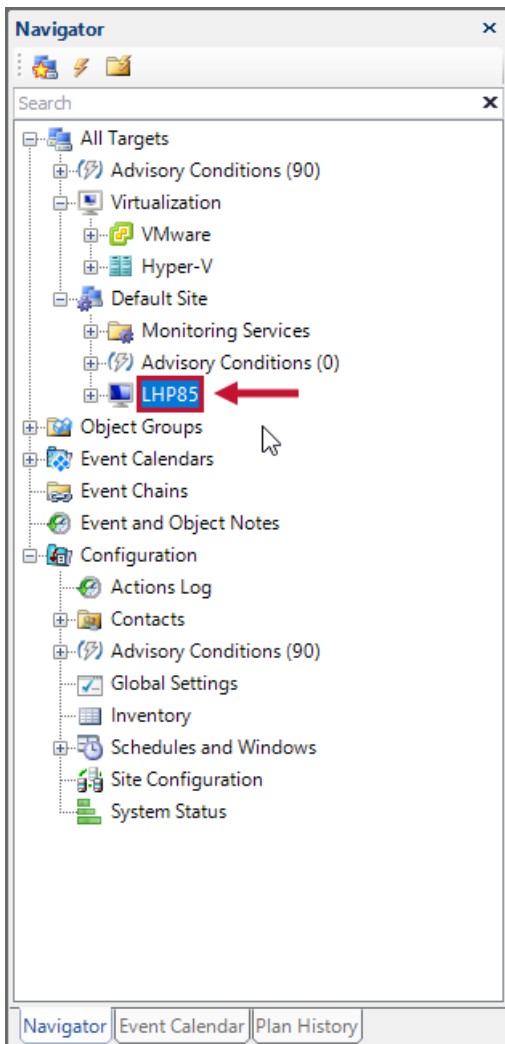
4. Change the **Minimum Block Duration** setting to **20** seconds, and then select **Ok**; it's saved automatically.



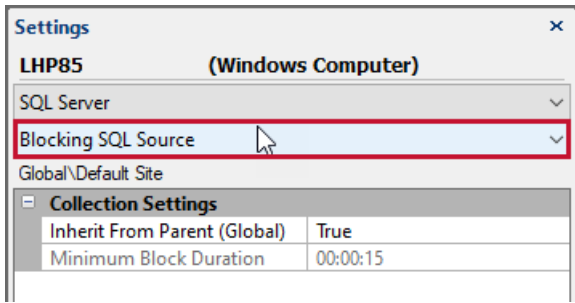
Example Two

Change the blocking collection for an individual instance by completing the following steps:

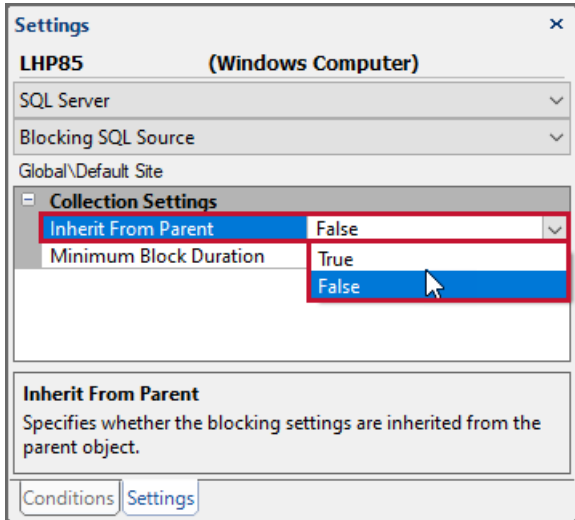
1. Select the desired instance node in the **Navigator** pane (**View > Navigator**).



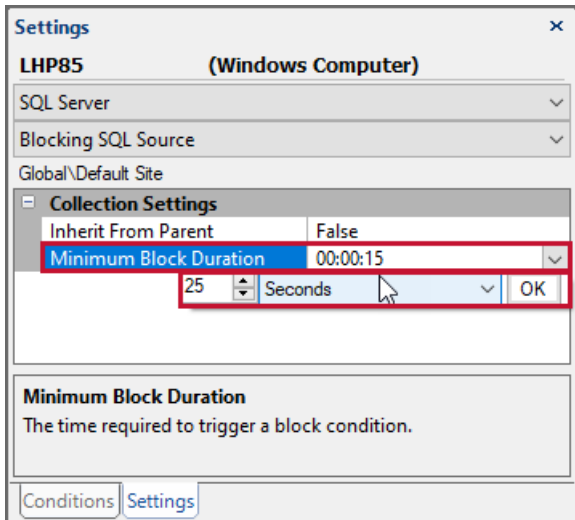
2. In the **Settings** pane, select **Blocking SQL Source** from the second drop-down list to see the **Blocking SQL Source** settings that are configured for the instance.



3. Change the **Inherit From Parent** setting to **False**.

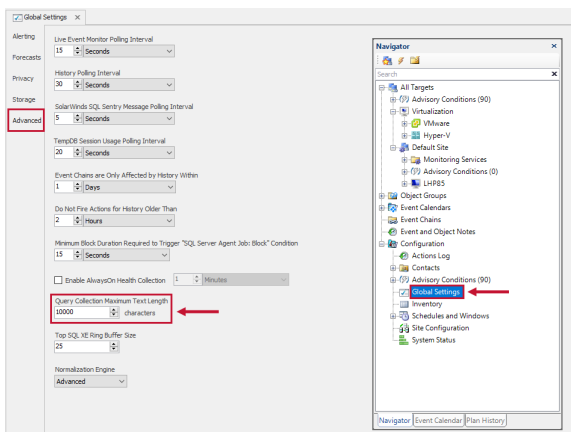


4. Change the **Minimum Block Duration** setting to your desired value; it's saved automatically.



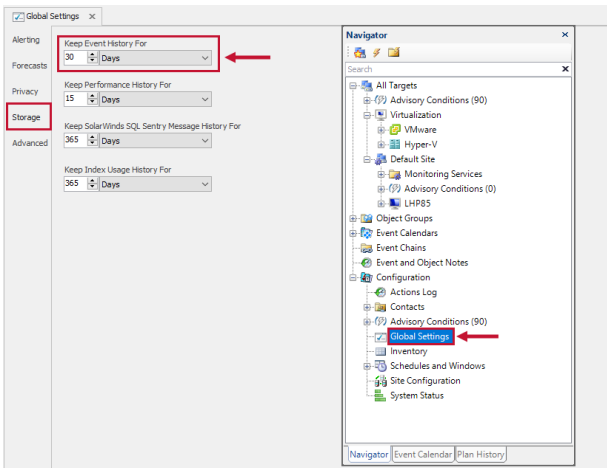
Query Collection Maximum Text Length

By default, SQL text data length for captured events is limited to 10,000 characters. Change this maximum length in the **Monitoring Service Settings (Navigator pane > Configuration > Global Settings > Monitoring Service Settings > Advanced tab > Query Collection Maximum Text Length)**.



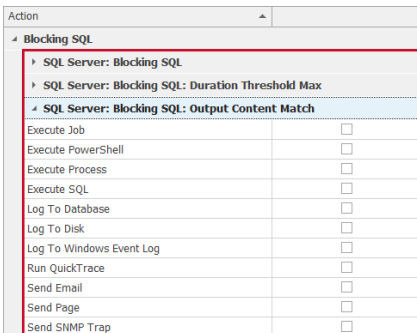
Blocking SQL Retention

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 **Global Settings (Navigator pane > Configuration > Global Settings > Storage tab)**. For more information about retention, see the [Data Capacity Planning](#) topic.



Blocking SQL Alerting

Configure actions in response to **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. Configure an action in response to the **SQL Server: Blocking SQL: Output Content Match** (OCM) condition. The **OCM** condition is useful in narrowing the notifications you see concerning blocking.

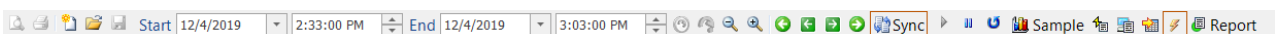


Note: Consider applying a Response Ruleset to any action you configure with the **SQL Server: Blocking SQL** condition. **Response Rulesets** control how often actions are taken in response to conditions. In this case, apply a **Response Ruleset** to limit the **Blocking** alerts you receive, while still retaining **Blocking** information on the **Blocking SQL** tab.

Display/Controls

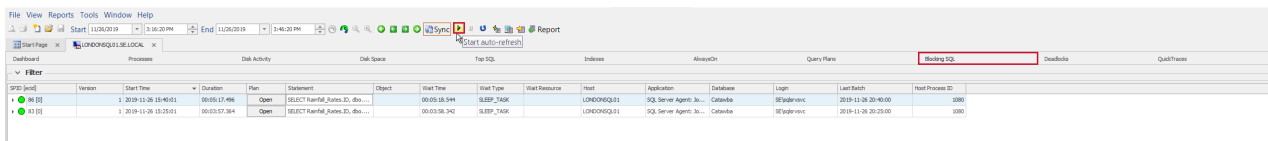
Toolbar

The **Blocking SQL** tab has two modes: **Real Time** and **History**. Use the toolbar options to switch between these modes.



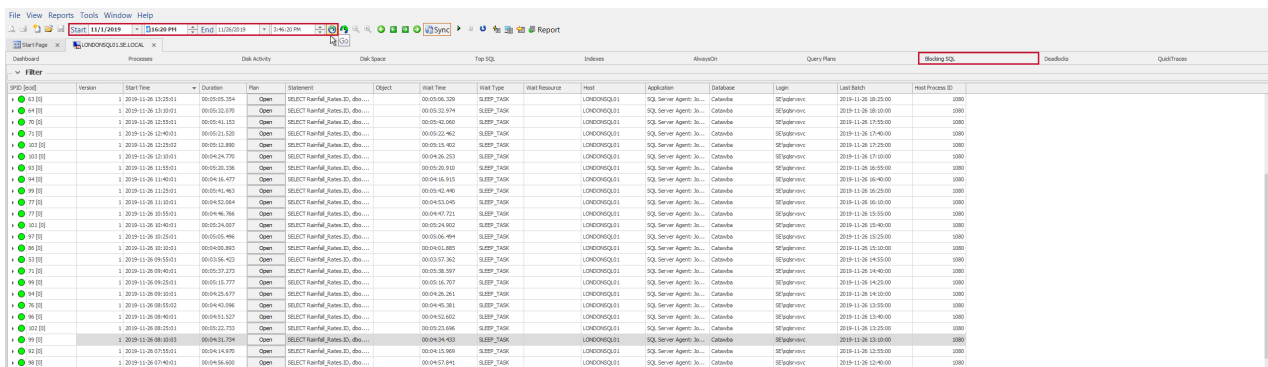
Real Time

When the auto-refresh **Play** button is selected, the tab enters **Real Time** mode. In **Real Time** mode, any blocking SQL that happened within the last 10 minutes displays.

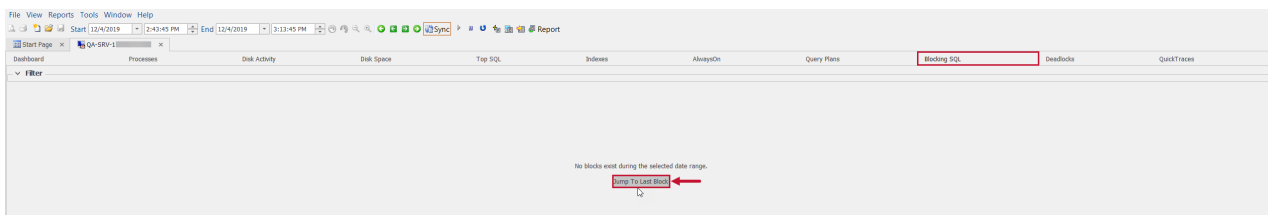


History

When the auto-refresh **Pause** button is selected, the tab goes into **History** mode. Select a time range, then select **Go** on the toolbar to view blocks that happened during the specified time range.

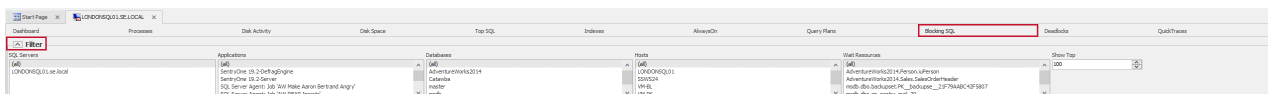


If you want to go directly to the most recent block, use the **Jump To Last Block** button.



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 only and doesn't impact what's collected. For more information, see the **Blocking SQL Collection** section.

After changing any of the filters, it's necessary to select **Refresh**, or press **F5**, to apply the filter. To select more than one item in a list, use **Shift-Click** or **CTRL-Click**. The following are available filters:

Filter

Image

| Filter | Image |
|----------------|-------|
| SQL Servers | |
| Applications | |
| Databases | |
| Hosts | |
| Wait Resources | |
| Show Top | |

Blocking SQL Grid View

Each block displays in a hierarchical format, showing the relationships between all blocking and blocked SPIDs in a blocking chain.

| SPID [col] | Version | Start Time | Duration | Plan | Statement | Object | Wait Time | Wait Type | Wait Resource | Host | Application | Database | Login | Last Batch | Host Process ID |
|------------|---------|---------------------|--------------|------|-----------------------------------|--------|--------------|------------|------------------|-------------|-------------------------|----------|-------------|---------------------|-----------------|
| 63 [G] | 1 | 2019-11-26 13:25:01 | 00:05:05.354 | Open | SELECT Rainfall_Rates.ID, dbo... | | 00:05:06.329 | SLEEP_TASK | | LONDONSQL01 | SQL Server Agent: 3a... | Catamba | SE\sqlservr | 2019-11-26 18:25:00 | 1080 |
| 64 [R] | 1 | 2019-11-26 13:20:01 | 00:09:32.070 | Open | SELECT Rainfall_Rates.ID, dbo... | | 00:05:32.974 | SLEEP_TASK | | LONDONSQL01 | SQL Server Agent: 3a... | Catamba | SE\sqlservr | 2019-11-26 18:10:00 | 1080 |
| 65 [R] | 1 | 2019-11-26 13:25:01 | 00:05:41.153 | Open | UPDATE [dbo].[Rainfall_Rates] ... | | 00:05:32.078 | LOCK_MU | RID: 101:1:168:0 | LONDONSQL01 | SQL Server Agent: 3a... | Catamba | SE\sqlservr | 2019-11-26 18:10:01 | 1080 |
| 79 [G] | 1 | 2019-11-26 13:25:01 | 00:05:41.153 | Open | SELECT Rainfall_Rates.ID, dbo... | | 00:05:42.060 | SLEEP_TASK | | LONDONSQL01 | SQL Server Agent: 3a... | Catamba | SE\sqlservr | 2019-11-26 17:55:00 | 1080 |

The head of the blocking chain is the top node in the hierarchy which contains the blocking statement, and displays with an orange node if the statement is still running, and a green node once it's completed.

| SPID [col] | Version | Start Time | Duration | Plan | Statement | Object | Wait Time | Wait Type | Wait Resource | Host | Application | Database | Login | Last Batch | Host Process ID |
|------------|---------|---------------------|--------------|------|----------------------------------|--------|--------------|------------|---------------|-------------|-------------------------|----------|-------------|---------------------|-----------------|
| 63 [G] | 1 | 2019-11-26 13:25:01 | 00:05:05.354 | Open | SELECT Rainfall_Rates.ID, dbo... | | 00:05:06.329 | SLEEP_TASK | | LONDONSQL01 | SQL Server Agent: 3a... | Catamba | SE\sqlservr | 2019-11-26 18:25:00 | 1080 |

Sub-nodes in the chain are the statements that are being blocked, and they display with a red node. Any subsequent statements that are being blocked are nested underneath, giving you a complete picture of the blocking chain.

| SPID [col] | Version | Start Time | Duration | Plan | Statement | Object | Wait Time | Wait Type | Wait Resource | Host | Application | Database | Login | Last Batch | Host Process ID |
|------------|---------|---------------------|--------------|------|-----------------------------------|--------|--------------|------------|------------------|-------------|-------------------------|----------|-------------|---------------------|-----------------|
| 63 [G] | 1 | 2019-11-26 13:25:01 | 00:05:05.354 | Open | SELECT Rainfall_Rates.ID, dbo... | | 00:05:06.329 | SLEEP_TASK | | LONDONSQL01 | SQL Server Agent: 3a... | Catamba | SE\sqlservr | 2019-11-26 18:25:00 | 1080 |
| 65 [R] | 1 | 2019-11-26 13:25:01 | 00:05:05.354 | Open | UPDATE [dbo].[Rainfall_Rates] ... | | 00:05:05.331 | LOCK_MU | RID: 101:1:168:0 | LONDONSQL01 | SQL Server Agent: 3a... | Catamba | SE\sqlservr | 2019-11-26 18:25:01 | 1080 |

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, meaning that a blocked SPID is either added or removed

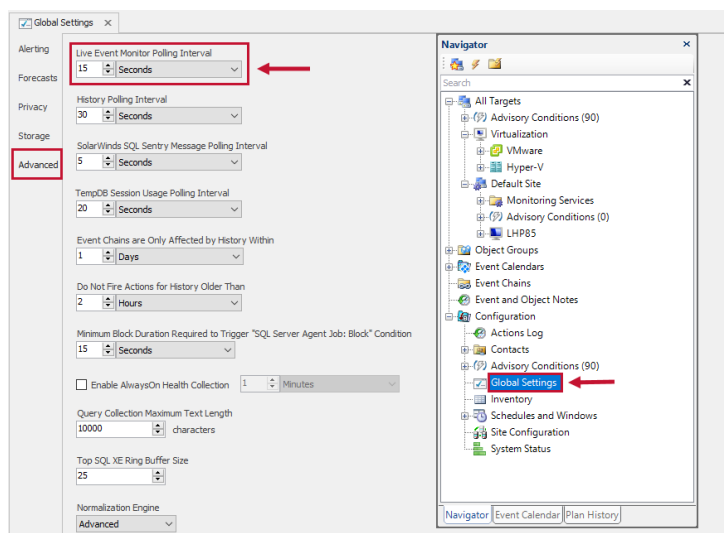
from the chain between polling intervals.

Note: 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.

Note: Indentations between SPIDs of the same color in the blocking chain indicate that the queries in the chain are being blocked by the currently blocked query.

| Color Code | Description |
|------------|---|
| Green | The head of the blocking chain is the top node in the hierarchy which contains the blocking statement and displays with a green node once it's completed. |
| Orange | The head of the blocking chain is the top node in the hierarchy which contains the blocking statement and displays with an orange node if the statement is still running. |
| Red | When sub-nodes in the chain are the statements that are being blocked they display with a red node. |

The polling interval for blocks is determined by the **Live Event Monitor Polling Interval (Navigator pane > Configuration > Global Settings > Monitoring Service Settings > Advanced tab).**



Text Data

The bottom pane shows the T-SQL **Text Data** for the currently highlighted Blocking SQL record.

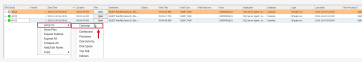
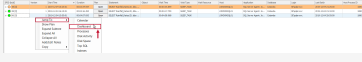
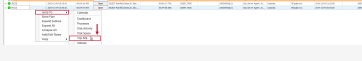
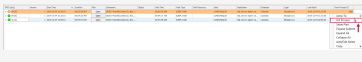

Text Data

```
SELECT Rainfall_Rates.ID, dbo.fnClrRetrieve(240684, 360562) AS Expr1
FROM Rainfall_Rates
WHERE Rainfall_Rates.ID = 10000
```

Note: For some SPIDs the bottom pane may be blank if SQL Sentry is unable to collect the TSQL due to timing issues.

Additional Options

The following are additional options available through the right-click context menu:

| Context Menu Option | Description | Depiction |
|-------------------------------------|--|---|
| Jump to Calendar | Opens the selected event on the Event Calendar (requires server to be watched with EM). |  |
| Jump to Dashboard | Opens the Performance Analysis Dashboard with the selected blocking event overlaid on each of the charts. |  |
| Jump to Top SQL | Opens the Top SQL tab highlighting the query involved in the blocking event. |  |
| Right-clicking on a blocking record | Presents a context menu that allows you to Jump To the instance on the calendar or kill the process associated with the blocking event. |  |
| Clear Sorting command | Removes any grid orderings. |  |

Additional context menu options include the ability to expand and collapse individual trees or all the records in the **Grid view**.

Blocking SQL Metrics

| Name | Description |
|--------------------|--|
| SPID [ecid] | The session process ID of the associated blocked/blocking process. |

| Name | Description |
|---------------|---|
| Version | <p>A unique record is shown for each version of a blocking chain, denoted by the Version column. A new record is created each time the blocking chain changes, meaning that a blocked SPID is either added or removed from the chain between polling intervals. For some blocks this may happen frequently, creating multiple records, while others may not change at all for the duration of the block.</p> <p>Note: The Polling Interval for blocks is determined by the Event Monitor Polling Interval (Navigator pane > Configuration > Global Settings > Monitoring Service Settings > Advanced tab).</p> |
| SQL Server | Name of the SQL Server hosting the SPID. |
| Start Time | Start time of the request. |
| Duration | The length of time that the block exists. |
| Plan | Opens a Plan Explorer session for the associated query plan. |
| Statement | The command text associated with the request. |
| Object | The object associated with the request. |
| Wait Time | Duration of wait time in milliseconds. |
| Wait Type | Name of the wait type. For more information about wait types, see the sys.dm_os_wait_stats (Transact-SQL) MSDN article. |
| Wait Resource | Name of the resource on which the request is currently waiting. |
| Host | The <u>client</u> workstation specific to this session. |
| Application | The associated application. |
| Database | The associated database. |
| Login | The login name associated with the session. |
| Last Batch | <p>The last time a parent process involved in blocking completed a successful batch execution.</p> <p>Note: This SPID metric can be useful if you have one that remains open and sends various batches. For example, if you are looking at SPIDs that are currently running, and see a SPID in a sleeping state, then you can see that the last batch execution was two hours ago.</p> |

| Name | Description |
|------------------------|---|
| Host Process ID | The process ID of the client program which initiated the session. |
