Consuming Events (Windows Event Log)

Article • 12/10/2020

You can consume events from channels or from log files. To consume events, you can consume all events or you can specify an XPath expression that identifies the events that you want to consume. To determine the elements and attributes of an event that you can use in your XPath expression, see Event Schema.

Windows Event Log supports a subset of XPath 1.0. For details on the limitations, see XPath 1.0 limitations.

The following examples show simple XPath expressions.

XML

```
// The following query selects all events from the channel or log file
XPath Query: *

// The following query selects all the LowOnMemory events from the channel or log file
XPath Query: *[UserData/LowOnMemory]

// The following query selects all events with a severity level of 1 (Critical) from the channel or log
file
XPath Query: *[System/Level=1]

// The following query shows a compound expression that selects all events from the channel or log file
// where the printer's name is MyPrinter and severity level is 1.
XPath Query: *[UserData/*/PrinterName="MyPrinter" and System/Level=1]

// The following query selects all events from the channel or log file where the severity level is
// less than or equal to 3 and the event occurred in the last 24 hour period.
XPath Query: *[System[(Level <= 3) and TimeCreated[timediff(@SystemTime) <= 86400000]]]
```

You can use the XPath expressions directly when calling the **EvtQuery** or **EvtSubscribe** functions or you can use a structured XML query that contains the XPath expression. For simple queries that query events from a single source, using an XPath expression is fine. If the XPath expression is a compound expression that contains more than 20 expressions or you are querying for events from multiple sources, then you must use a structured XML query. For details on the elements of a structured XML query, see Query Schema.

A structured query identifies the source of the events and one or more selectors or suppressors. A selector contains an XPath expressions that selects events from the source and a suppressor contains an XPath expression that prevents events from being selected. You can select events from more than one source. If a selector and suppressor identify the same event, the event is not included in the result.

The following shows a structured XML query that specifies a set of selectors and suppressors.

XML

```
<QueryList>
  <Query Id="0">
    <Select Path="Application">
        *[System[(Level <= 3) and
        TimeCreated[timediff(@SystemTime) <= 86400000]]]
    </Select>
    <Suppress Path="Application">
        *[System[(Level = 2)]]
    </Suppress>
```

```
    <Select Path="System">
        *[System[(Level=1  or Level=2 or Level=3) and
        TimeCreated[timediff(@SystemTime) <= 86400000]]]
    </Select>
  </Query>
</QueryList>
```

The result set from the query does not contain a snapshot of the events at the time of the query. Instead, the result set includes the events at the time of the query and will also contain all new events that are raised that match the query criteria while you are enumerating the results.

> **Note**
>
> The order of the events is preserved for events that are written by the same thread. However, it is possible for events written by separate threads on different processors of a multiple processor computer to appear out of order.

For details on consuming events, see the following topics:

- Querying for Events
- Subscribing to Events
- Rendering Events
- Formatting Event Messages
- Bookmarking Events

The standard end user tools for consuming event are:

- Event Viewer
- The Windows PowerShell Get-WinEvent cmdlet
- **WevtUtil**

XPath 1.0 limitations

Windows Event Log supports a subset of XPath 1.0. The primary restriction is that only XML elements that represent events can be selected by an event selector. An XPath query that does not select an event is not valid. All valid selector paths start with * or "Event". All location paths operate on the event nodes and are composed of a series of steps. Each step is a structure of three parts: the axis, node test, and predicate. For more information about these parts and about XPath 1.0, see XML Path Language (XPath) . Windows Event Log places the following restrictions on the expression:

- Axis: Only the Child (default) and Attribute (and its shorthand "@") axis are supported.
- Node Tests: Only node names and NCName tests are supported. The "*" character, which selects any character, is supported.
- Predicates: Any valid XPath expression is acceptable if the location paths conform to the following restrictions:
  - Standard operators **OR**, **AND**, =, !=, <=, <, >=, >, and parentheses are supported.
  - Generating a string value for a node name is not supported.
  - Evaluation in reverse order is not supported.
  - Node sets are not supported.

- Namespace scoping is not supported.
- Namespace, processing, and comment nodes are not supported.
- Context size is not supported.
- Variable bindings are not supported.
- The position function, and its shorthand array reference, is supported (on leaf nodes only).
- The Band function is supported. The function performs a bitwise AND for two integer number arguments. If the result of the bitwise AND is nonzero, the function evaluates to true; otherwise, the function evaluates to false.
- The timediff function is supported. The function computes the difference between the second argument and the first argument. One of the arguments must be a literal number. The arguments must use FILETIME representation. The result is the number of milliseconds between the two times. The result is positive if the second argument represents a later time; otherwise, it is negative. When the second argument is not provided, the current system time is used.

Feedback