**Retrieving Wonderware Historian Alarm & Event Data through OData Clients:**

Alarms & Events can be retrieved from Wonderware Historian 2014 R2 or 2014 R2 patch 01 using following OData Clients:

1. Postman-Rest Client
2. Power Query on MS Excel 2013
3. Power Pivot on MS Excel 2013

**Prerequisites for testing alarm data:**

* Run the Vijeo Historian Connector v2.0 to push the alarm & events from Vijeo Citect SCADA to Wonderware Historian.
* Make sure that the Vijeo Citect alarms & events are historized in Historian.
* In WonderWare Historian system, add windows logged in user or Historian user to windows groups like aaAdministrators or aaPowerUsers.

**Below are the steps to use each of these Clients**

**Postman-Rest Client:**

1. Download the App Postman-Rest Client from Chrome web store.
2. Launch the App.
3. Enter the following URL in the text box named “*Enter request URL here”.*

*http://<WWH System name>:32569/Historian/v1/Events*

1. When prompted for username and password, provide the Wonderware Historian user credentials (can be windows logged in user).
2. User would be part of **aaAdministrators** or **aaPowerUsers**.
3. Select the **Get Request** and click **Send** button.
4. The response will be displayed with each alarm and its properties.
5. Check following section 'Supported System Query Options for OData Event Retrieval' in Wonderware Historian 2014 R2 Help to try the request with different tokens and logical operators.
6. Try data with different filter and select functions along with logical operators and verify the results.

**Example:** *http://localhost:32569/Historian/v1/Events?$filter=EventTime gtdatetimeoffset'2014- 10- 28T06:52:10.081Z'*

1. Also can use the same query on a browser, get the JSON file and open it in notepad to verify the response.

**Power Query in MS Excel 2013:**

1. Install MS Excel 2013.
2. Download and Install Power Query (*http://www.microsoft.com/en-us/download/details.aspx?id=39379*).
3. Open Excel and select **Power Query** tab.
4. Select **GetExternalData** and select option **From Other Sources**.
5. Select **From ODatd Feed**.
6. OData Feed dialog appears and provide the below URL and click **OK** button:

*http://localhost:32569/Historian/v1/Events*

1. Select **Authorization** tab and provide logged in windows user credentials.
2. The data will be displayed in tabular format with all alarm properties in different columns.
3. Use Power Query options to format and view relevant data.

**Power Pivot in MS Excel 2013:**

1. Install MS Excel 2013.
2. Enable **Power Pivot** using Excel Options - **Add Ins**.
3. Launch Power Pivot.
4. Select **Manage** button **→** **Get External Data** **→ From Data Service** **→ From OData Feed**.
5. Provide a Friendly Connection Name and provide following URL in Data Feed.

*http://localhost:32569/Historian/v1/Events?$format=atom*

1. Test the connection and click **Finish**.
2. Load the data to table in existing worksheet.
3. View the data and change the URL with different tokens and logical operators to query different data set.

**The following table describes the supported system query options:**

**Note:** The options are case-sensitive.

|  |  |
| --- | --- |
| **Token** | **Description** |
| $filter | Specifies an expression or a function that must evaluate to true for a record to be returned in the collection.  All typical OData functions are supported for the $filter clause.  The $filter expression supports references to properties and literals. Literal values include:   |  | | --- | | * Strings enclosed in single quotes | | * Numbers and Boolean values (true or false) | |  |  | |  |  |   While event property names are case-sensitive and filtering is case-insensitive. For example, if you filter property values based on a value of "true", values such as "TRUE", "True," and "true" could be returned. The case returned in the query results reflects the case of the stored value. |
| $select | Specifies a subset of properties to return. |
| $skip | Specifies the number of records to skip from the beginning of the result set. |
| $skiptoken | Used to get the next record set that satisfies the query conditions. You do not need to include this token in the query, but you will see it upon query execution. |
| $top | Specifies the maximum number of records to return. This subset is formed by selecting only the first **N** items of the set, where **N** is a positive integer specified by this query option. |

**The following table describes supported logical operators for the query options:**

|  |  |
| --- | --- |
| **Logical Operator** | **Description** |
| eq | Equal |
| ne | Not equal |
| gt | Greater than |
| ge | Greater than or equal |
| lt | Less than |
| le | Less than or equal |
| and | Logical and |
| or | Logical or |
| not | Logical negation |

**OData Query Examples**

**The following query returns events after the specified date:**

http://Server1:32569/Historian/v1/Events?$filter=EventTime ge datetimeoffset'2014-08-10T05:56:21.302Z'

**The following query returns events for the process variable named Flow3.PV:**

http://Server1:32569/Historian/v1/Events?$filter=Source\_ProcessVariable eq 'Flow3.PV' and EventTime ge datetimeoffset'2014-08-10T05:56:21.302Z'

**The following query returns events for a source area called “Site2”:**

http://Server1:32569/Historian/v1/Events?$filter=Source\_Area eq 'Site2' and EventTime ge datetimeoffset'2014-08-10T05:56:21.302Z'

**The following query returns alarms that have returned to normal:**

http://Server1:32569/Historian/v1/Events?$filter=Type eq 'Alarm.Clear'

**The following query returns alarms that have returned to normal and uses the $select option to return only a subset of the properties:**

http://Server1:32569/Historian/v1/Events?$select=EventTime,Source\_Area,Source\_ProcessVariable,Comment&$filter=Type eq 'Alarm.Clear'

**The following query returns all events with an event time that is greater than or equal to the specified time:**

http://Server1:32569/Historian/v1/Events?$filter=EventTime ge datetimeoffset'1970-01-01T00:00:00.000Z'

**The following query returns the first 10 events:**

http://Server1:32569/Historian/v1/Events?$top=10&$filter=EventTime ge datetimeoffset'1970-01-01T00:00:00.000Z'

**The following query skips the first 10 events:**

http://Server1:32569/Historian/v1/Events?$skip=10&$filter=EventTime ge datetimeoffset'1970-01-01T00:00:00.000Z'

**Handling Alarms at Runtime**

Once you have configured your alarms you then need to create alarm pages so operators can view and acknowledge alarms within your system. Using a tab style template or configuring a standard graphics page you can create the following alarm pages:

|  |  |
| --- | --- |
| **AlarmPage** | **Description** |
| Active | When an alarm is triggered, it becomes active and is displayed on the Active Alarm page. The active state of a digital alarm is ON, while the active state of an analog alarm varies depending on the type of alarm (for instance HIGH, LOW , RATE, and so on).  The operator can acknowledge every alarm on page by clicking **Alarm Ack**.  When an alarm is acknowledged, it displays change in color, and its state changes to ACKNOWLEDGED. When the alarm is reset (when the conditions that caused the alarm have been rectified), its state changes to OFF. When the state changes to OFF the Alarm is no longer displayed and is available from the Disabled Alarm page. You can then view state changes (events) for that alarm using either the Alarm SOE page. |
| Disabled | If an alarm does not appear to be operating as designed, or is deemed unnecessary, an operator can disable it. Disabled alarms are ignored until they are re-enabled.  (you need to define a command that uses the **AlarmDisable()** function to disable alarms.) |
| SOE | To maintain a history of alarm activity, an event log is stored. This log stores the time when each alarm was activated, acknowledged, and reset. You can display alarms from the event log (including disabled alarms) on the sequence of events and alarm summary page. The SOE page displays actions taken on an alarm and user events such as login. You can also add comments to existing events and insert new events into the event journal. Manually refresh the page to view the latest events.  **Note**: In v7.40 Vijeo Citect alarm login system requires two levels of login:   * Login to the Client system supported by the “Login successful” prompt-message. * Login to the Alarm DB Server visible via SOE events.   Successful Alarm DB login events are viewable via the SOE page. The login events can be filtered and sorted by date-time and other filter criteria, with the specific login event identified by the “Logged on” entry in the Message column and the corresponding login name in the “UserName” column.  Unsuccessful DB login/connection events though not triggering new SOE entries, may generate “DB not connected” Hardware alarms. |
| Summary | The Alarm Summary page displays ON/OFF Alarms, acknowledged and unacknowledged states. Manually refresh the summary page to view the latest events before analyzing the data. |
| Hardware | Displays details of any system errors that are unacknowledged/acknowledged and still in an alarm state. |

**To display an alarm page at runtime:**

1. Create an alarm (or hardware alarm) page in your project if you have not already done so. Call the **Alarm** page for a configurable alarm page and **Hardware** page for a hardware alarm page.
2. Create a new keyboard command or a button, to call the page at runtime. You can also add a touch command to an existing screen object.
3. In the command field, enter **PageAlarm()** to display the configurable alarms page, or **PageHardware()** to display the hardware alarms page.
4. Configure other properties as necessary.
5. Click **Add** to append a new record, or **Replace** to modify an existing record.

**Note:** If using the standard page templates, you don't usually need to create a command to display the page as the commands are already built in.

To display a customized alarm page (with a non-standard name), use the **PageDisplay()** function to display the page, followed by the **AlarmSetInfo()** function as necessary.