M.A.C. SOLUTIONS

PROCESSVUE

Alert to ProcessVue Application Note

Abstract

How to retrieve Alert Data and record it within ProcessVue using SQL triggers. This document will demonstrate the steps needed for ProcessVue using SQL triggers and the Table Monitor functionality within the ProcessVue Data Collector

Contents

1. Introduction 3
1.1 Background Knowledge 3
1.2 Solution Descriptive Diagram 3
2. How to Connect Alert to ProcessVue 4
2.1 Configuration 4
2.2 Linking Alert with ProcessVue Steps 4
3. End Result 7

1. Introduction

This document will show the steps needed to retrieve Alert's data which is stored in Microsoft SQL server into a ProcessVue friendly database using SQL triggers. Alert's data can be stored within a Microsoft Access Database and the data can be passed into SQL server using an ODBC connection.

1.1 Background Knowledge

Alert is designed to insert a new record into the database when an alarm is flagged and then updates the record with the acknowledge time and the reset time. This approach means that three events will be stored in one line, e.g.

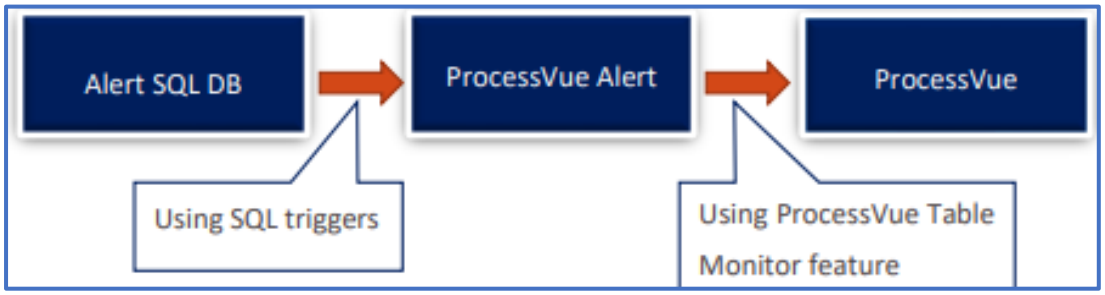
Alarm Log Id	Tag Id	Message	Alarm Time	Ack Time	Reset Time
1	1	Alarm test 001	2017-06-29 10:04:38.077	2017-06-29 10:09:48.447	2017-06-29 11:04:43.343

ProcessVue table monitor is designed to read all events in a chronological order using an event ID and since alert records three events (alarm, acknowledge and return to normal) in one line, an alternative had to be developed to allow ProcessVue table monitor to read the data correctly. The developed solution is designed to create a unique entry in a database for every event using SQL triggers, the above tables will be translated into the table below.

Event ID	Event Date Time	Tag ID	Alarm Action
1	2017-06-29 10:04:38.077	1	ALM
2	2017-06-29 10:09:48.447	1	ACK
3	2017-06-29 11:04:43.343	1	RTN

1.2 Solution Descriptive Diagram

The following diagram illustrates the data flow of our solution from the Alert in ProcessVue.



2. How to Connect Alert to ProcessVue

This section will detail the configuration needed to get data from an alert system into ProcessVue followed by the steps needed to get this fully operational. To do this, we need the following:

1. An Alert system with an ODBC connection to SQL server

2. SQL Server with management studio

3. The SQL scripts needed to create the triggers

4. ProcessVue system

2.1 Configuration

For the ProcessVue table monitor to work correctly, alert's data has to be saved in a chronological table with unique id for each record. Therefore, repository database and table have to be created. The table then needs to be populated using SQL triggers. The SQL scripts needed are as follows:

1. Create_DB_&_Table.sql: This script creates the repository database and table; the database is called 'ProcessVueAlert' and the table is called 'ProcessVueLog' which contains a column called 'Alarm Action' which defines the event type.

2. Create_ALM_Trigger.sql: This creates the 'OnInsertTrigger' trigger which adds a record with alarm action 'ALM' to the repository table when a new record is added to the alert database.

3. Create_ACK_Trigger.sql: This creates the 'OnUpdateTriggerACK' trigger which adds a record with alarm action 'ACK' to the repository table when the alert record gets updated with a new acknowledgement time.

4. Create_RTN_Trigger.sql: This creates the 'OnUpdateTriggerRTN' trigger which adds a record with alarm action 'RTN' to the repository table when the alert record gets updated with a new reset time.

2.2 Linking Alert with ProcessVue Steps

Step 1: Install Alert and set it to write to database and link the access database to SQL server using ODBC. Alert SQL database should be called 'AlertSQL_DB' as the trigger scripts will reference it.

Step 2: Run the create triggers scripts in the following order:

1. Create_DB_&_Table.sql

2. Create_ALM_Trigger.sql 4

3. Create_ACK_Trigger.sql

4. Create_RTN_Trigger.sql

Step 3: Install and licence the ProcessVue system which consist of the followings:

1. ProcessVue Data Collector & Configurator

2. ProcessVue Data Archiver & Configurator

3. ProcessVue Data Analyser & Configurator *

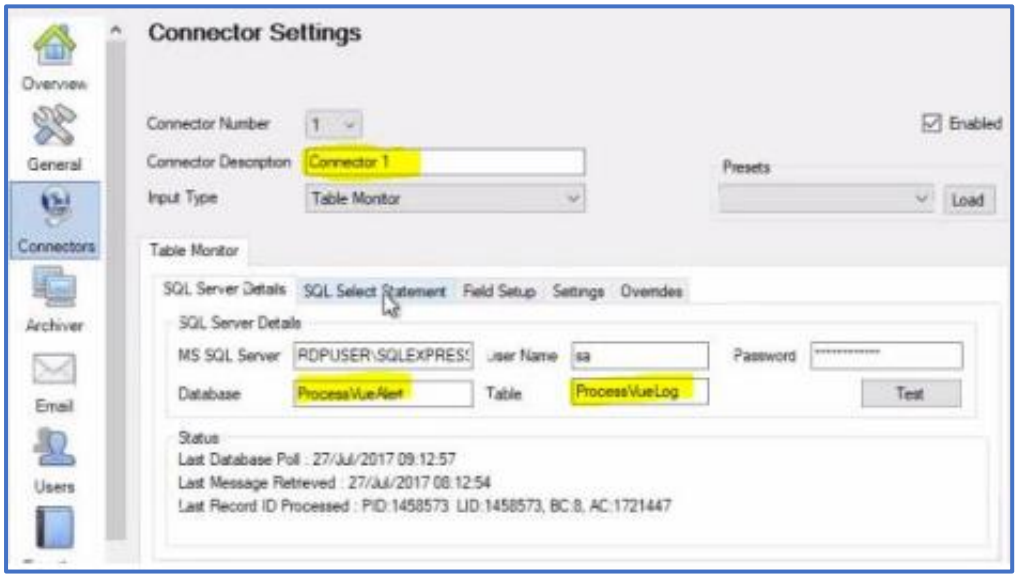
4. ProcessVue SOE Web application *

5. ProcessVue Analyser Web application *

* Available on selected packages.

Step 4: Configure the Data Collector Table monitor to retrieve the data from the repository table as follows:

1. Set the SQL Server connection

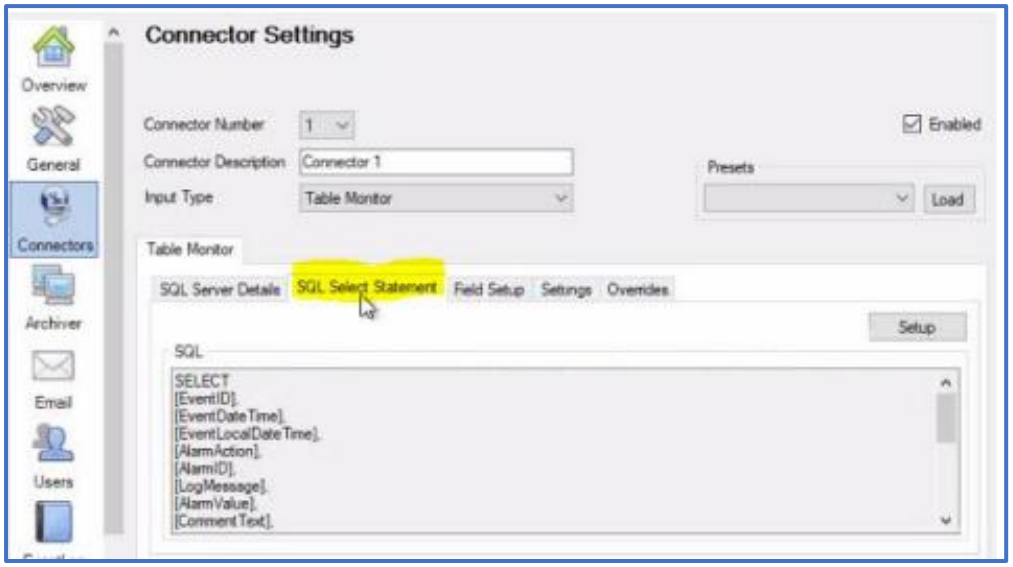


2. Configure the below SQL Select Statement:

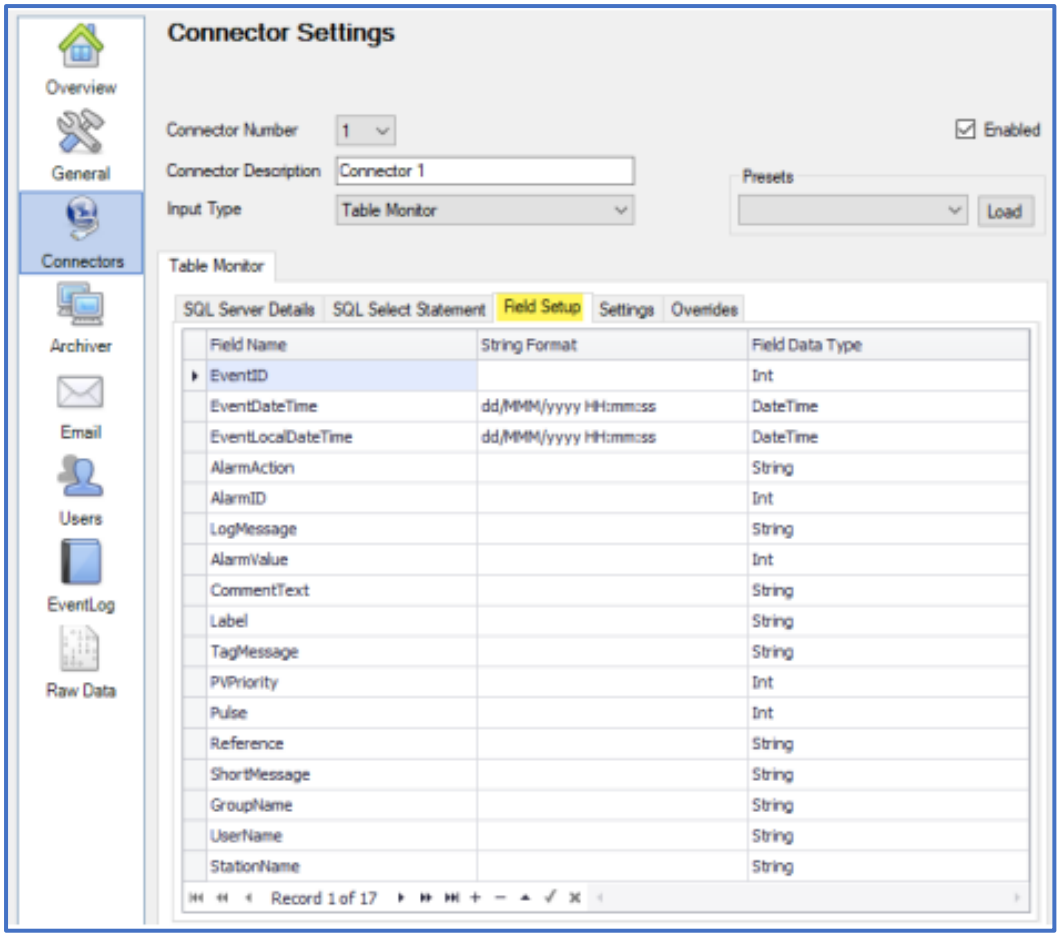
SELECT

[EventID], [EventDateTime], [EventLocalDateTime], [AlarmAction], [AlarmID], [LogMessage], [AlarmValue], [CommentText], [Label], [TagMessage], [PVPriority], [Pulse], [Reference], [ShortMessage], [GroupName], [UserName], [StationName]

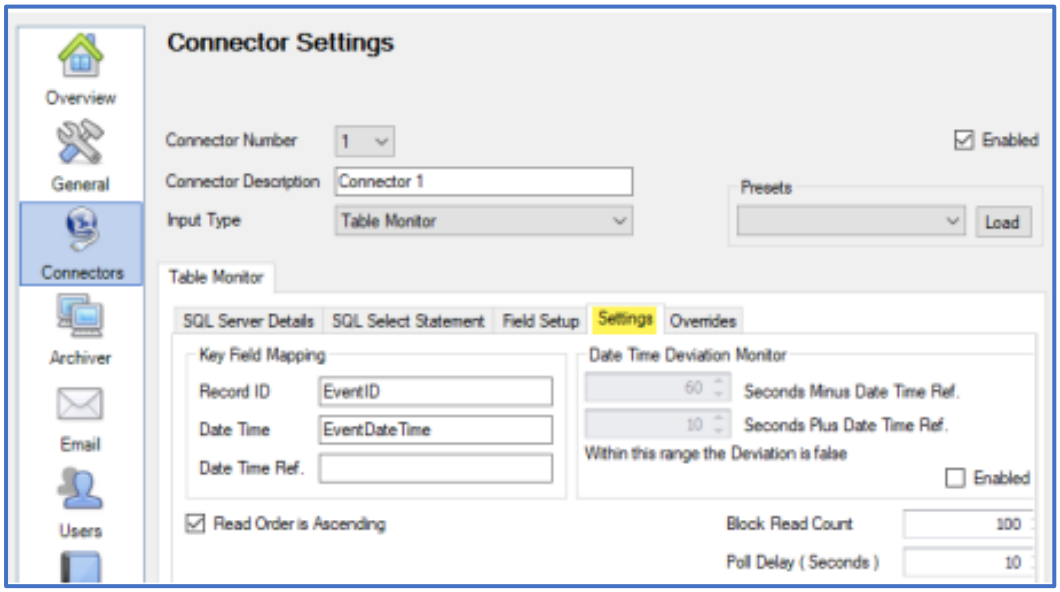
FROM [ProcessVueLog]



3. Configure the fields as follows:

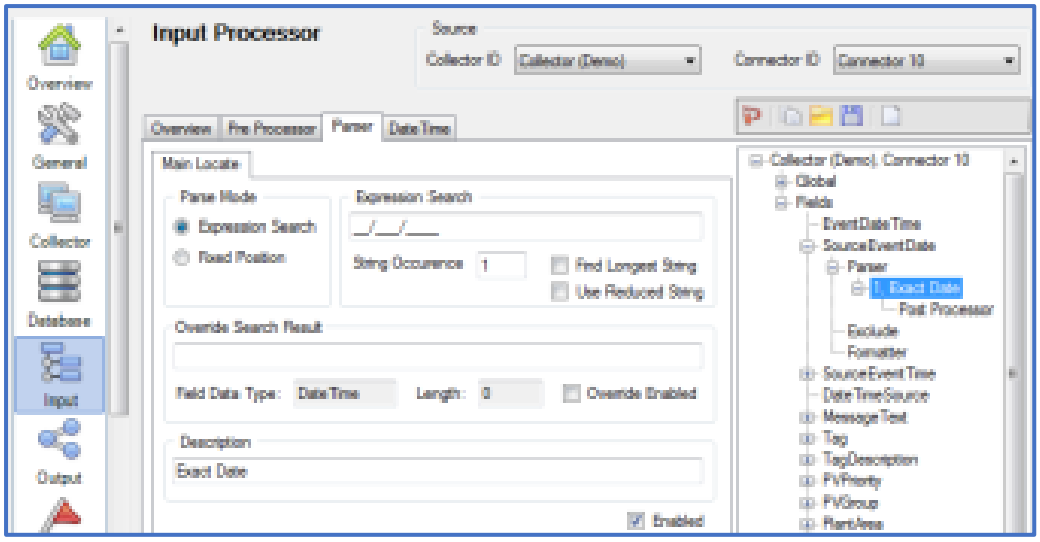


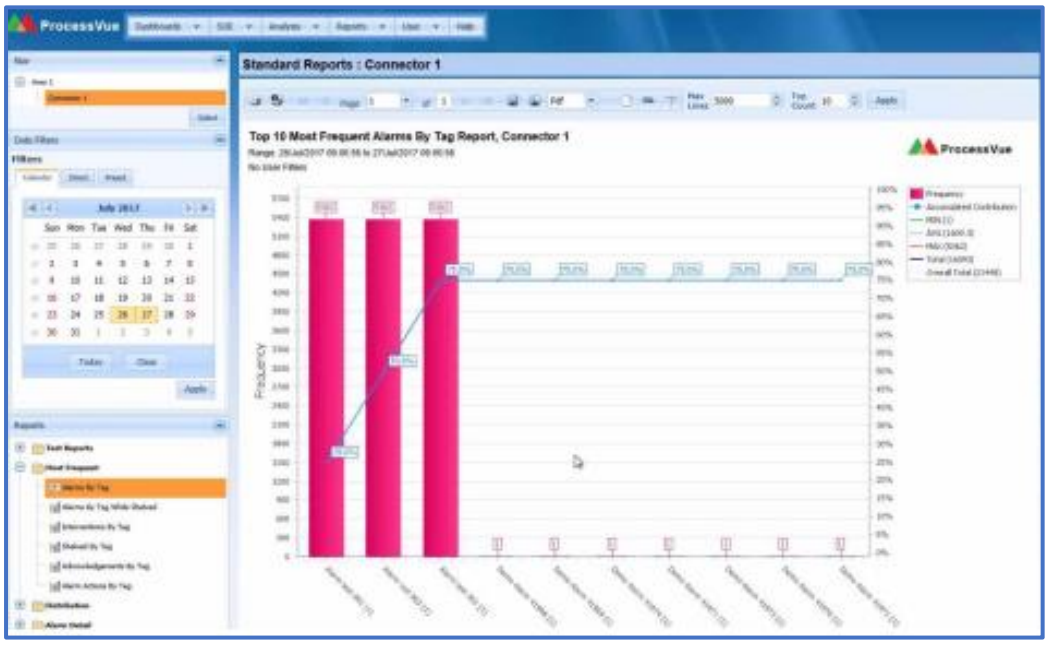
4. Configure the Read Settings



5. Configure the Data Archiver parsing rules:

Each connector/input is logged to a table in the MS SQL Server ProcessVue Database. Each table contains a list of User-definable fields. ProcessVue uses the input Process Parser to extract the data into the Fields. More details on Parsing are included in the Data Archiver help files.



3. End Result

1,455,008 messages logged by Alert have been recorded correctly to ProcessVue by having a script within Alert that sends tags into alarm, then 5 seconds later into acknowledge, and 5 seconds later it goes into normal. The chart below demonstrates the Most Frequent Alarm by Tag for a duration of one day.