

Export lista allarmi da Alarm Client in file .csv

Introduzione

Questa TN mostra come esportare la lista allarmi tramite script, accedendo alle proprietà dell'Alarm Client – quindi valido sia per storicizzazione con Alarm DB Logger (WWALMDB) che con Historian. Le procedure descritte porteranno a un export in un file .csv

Versioni

Quanto descritto in questa TN è stato verificato sulle seguenti versioni:

- Dalla versione 2017 U3 SP1 in avanti

Export della Lista Allarmi dall'Alarm Client

E' possibile quindi esportare anche la lista degli **allarmi attivi**, che esportare la lista degli **Allarmi Storici**. Visto che accediamo direttamente all'Alarm Client, con lo stesso metodo è possibile estrarre gli allarmi storicizzati sia da **WWALMDB**, che da **Historian**.

Si può esportare la lista allarmi direttamente da quanto è visualizzato sull'Alarm Client di Archestra, accedendo alle sue proprietà tramite scripting.

Nel codice si assume che il nome dell'alarm client nel simbolo sia "AlarmClient", che il separatore voluto sia la virgola semplice, che la directory di salvataggio sia la root del disco C: e il nome del file di output sia "AlarmExport.csv".

Le colonne vengono considerate solo se visibili nell'Alarm Client.

Tramite replace, viene inoltre fatto l'escape dei doppi apici, in modo che, ad esempio, la stringa:

Allarme "Filtro 3" bloccato, contattare assistenza

nel file CSV venga resa così:

"Allarme ""Filtro 3"" bloccato, contattare assistenza"

come di solito ci si aspetta, per evitare problemi con i campi che possono contenere doppi apici e virgole in file di valori separati da virgola.

1. Procediamo quindi a Embeddare un Alarm Client in un simbolo Archestra (nominato AlarmClient) e configurarlo in Historical Mode (nel nostro caso, storicizziamo gli allarmi su Historian);
2. Creiamo una Custom Property booleana **ExportAlarms** per scatenare lo script, un button che setta la Custom Property Trigger a True;
3. Premiamo F10 e creiamo uno script nominato 01_AlarmExportCSV e come condizione mettiamo **ExportAlarms = true** e come modalità mettiamo all' **OnTrue**;

4. Inseriamo lo script seguente per esportare il file AlarmExport.csv nella cartella

C:\AlarmExport\:

```
dim separator = ",";
dim path = "C:\AlarmExport\AlarmExport.csv";

dim csv = "";
dim i as integer;
dim j as integer;

ExportAlarms = false;

for i = 0 to alarmclient.ColumnsDetailEntity.ColumnList.Count - 1
    if alarmclient.ColumnsDetailEntity.ColumnList[i].IsVisible then
        if i > 0 then csv = csv + separator; endif;
        csv = csv + """" +
        alarmclient.ColumnsDetailEntity.ColumnList[i].DisplayName + """";
    endif;
next;
csv = csv + System.Environment.NewLine;

for i = 0 to AlarmClient.RowCount - 1
    for j = 0 to AlarmClient.ColumnsDetailEntity.ColumnList.Count - 1
        if AlarmClient.ColumnsDetailEntity.ColumnList[j].IsVisible then
            if j > 0 then csv = csv + separator; endif;
            csv = csv + """" + AlarmClient.GetItem(
                i,
                AlarmClient.ColumnsDetailEntity.ColumnList[j].ColumnName
            ).Replace("""", """"""") + """";
        endif;
    next;
    csv = csv + System.Environment.NewLine;
next;

System.IO.File.WriteAllText(path, csv);
```

5. In questo modo, verrà esportata la lista degli allarmi visualizzati sull'Alarm Client.

N.B. Da notare che lo script sopra fornito è solo a titolo di esempio, vanno quindi considerati aspetti quali:

- l'esistenza del file (e del percorso);
- rendere dinamico percorso / nome file all'occorrenza, ad esempio aggiungendo data-ora di generazione al nome del file per evitare sovrascritture
- timeout di esecuzione: assicurarsi che l'alarm client non visualizzi troppi record o gestire di conseguenza lo script affinché non vada in timeout.

Referenze

AVEVA™ Alarm Client Control: <https://docs.aveva.com/bundle/sp-alarm/page/840468.html>