

Description: Guide to configure the cross-referencing table(s) necessary to allow analytical reports from non-standard AlarmWorX32 Logger database formats.

OS Requirement: Win XP/Server 2003/ Server 2008/Vista/Windows 7

General Requirement: BizViz V9.2x and general knowledge of Alarm Analytics and the AlarmWorX32 Logger.

Introduction

This application note details the configuration requirements necessary to allow BizViz V9.2x Alarm Analytics to analyze alarm data collected by older AlarmWorX32 versions as well as third party logged historical alarm databases.

As released, the V9.2x Alarm Analytics queries assume the latest mapping schema of OPC Alarm and Event (OPC A/E) attributes to the native AlarmWorX32 alarm historical database table columns. This presents a compatibility issue with respect to older historical alarm databases logged by AlarmWorX32 or third party historical alarm loggers.

NOTE: This application note serves as a supplement to the ICONICS *Alarm Analytics - Setup Wizards Provide Industry Standard Reports* Applications Note.

Default Use of AWX_LookupMap

In the V9 release, the Alarm Analytics provider relies on the AWX_LookupMap table, which contains all aliasing information for mapping OPC A/E alarm attributes to the native columns that the AlarmWorX32 historical alarm logger use.

AWX_LookupMap : Table				
	ChangeMask	Newstate	Source	EventTime
	ChangeMask		Tag	EventTime
	*			Messa

Figure 1 - Alarm Attribute Mapping in Default AlarmWorX32 Logging Database

This presents a third party compatibility issue since the Alarm Analytics provider cannot rely on the existence of this table in the source alarm historical database. In addition, even if the AWX_LookupMap table is present, the end user may not be able to modify its structure because of FDA or security related validation constraints.

The Compatibility Enhancement

To enable the Alarm Analytics provider to be compatible with older or third party alarm databases, an intermediate translation

layer needs to be created, which will allow the end user to define custom mappings of the source historical alarm database to the historical alarm table column schema that the BizViz Alarm Analytics provider is designed upon.

The enhancement introduces the utilization of a local MS Access (.mdb) file, which provides the compatibility mappings for the BizViz Alarm analytics provider to interface to historical alarm databases with different column structures. Once this information is stored, the end user of the BizViz Alarm Analytics provider will be able to configure and execute Alarm Analytics based reports

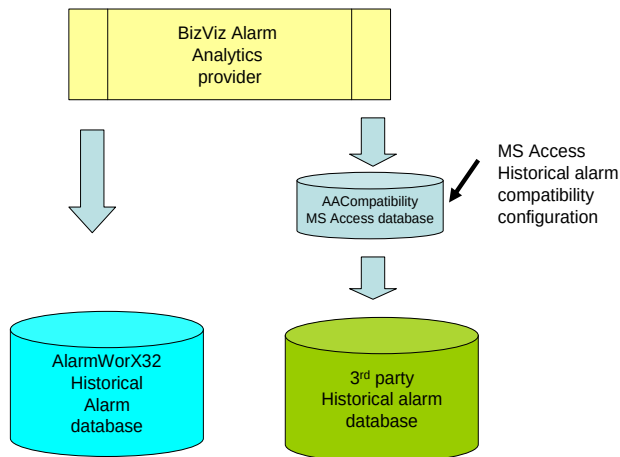


Figure 2 - Alarm Analytics Compatibility Architecture

The alarm compatibility database provides the following benefits:

- Configuration of compatibility mappings for several historical alarm databases
- Central management of these configurations
- Support for database expressions to compensate for required but missing source historical alarm table configurations
- Easy integration approach by OEM's to support historical alarm compatibility column mappings

Editing the AACompatibility Database

The AACompatibility Database hosts the necessary information to translate database format that Alarm Analytics does not understand to one that it can work with. Figure 3 shows the different tables in this database.

In order to for Alarm Analytics to understand virtually any alarm attribute column naming, you will have to add relevant information about your original historical alarm database. We will describe all the necessary steps in this section.

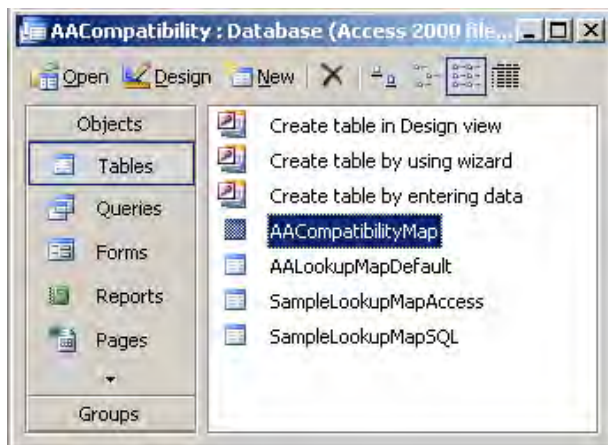


Figure 3 - Alarm Analytics Compatibility Database

Lookup Map Table

A Lookup Map table tells Alarm Analytics what columns it should look for in the alarm log database when it is looking for certain information. For example, many AlarmWorX32 Logging applications name the source column “Tag”. In order for Alarm Analytics to know that the source information is in the Tag column it needs to look it up in a Lookup Map table.

In a version 9 AlarmWorX32 Logging database there is a table called AWX_LookupMap that provides this information. When there is no AWX_LookupMap available we can use the AALookupMap database to replace it.

A Lookup Map table has five columns:

- *KeyID* – A unique number that identifies each lookup item
- *AAWX_LookupMap_Name* – The name of the alarm attribute that Alarm Analytics may look for
- *AAWX_LookupMap_Actual_Name* – The name of the column where this information resides in your alarm log database
- *RequiresCalculation* – If this box is checked then this alarm attribute does not exist natively in the alarm log database and it must be calculated from multiple columns
- *Expression* – The expression used to calculate this attribute if the RequiresCalculation box is checked

The first two columns, KeyID and AAWX_LookupMapName should not be altered, but you may need to edit the other three columns to match your alarm log database schema.

In some cases the default or sample Lookup Map tables can be used unedited, but if you are using a third party alarm database or have a highly customized AlarmWorX32 Logger database from an older version, you may need to follow these steps to

create your own Lookup Map table. A brief description of each of the sample Lookup Map tables is below.

- AALookupMapDefault – The default Lookup Map
- SampleLookupMapAccess – An example Lookup Map that might be used by an Access database.
- SampleLookupMapSQL – An example Lookup Map that might be used by a SQL Server database.

To make a Lookup Map table for your custom database, follow these steps:

1. Browse to your ReportWorX Service directory (which is by default, *C:\Program Files\ICONICS\BizViz\ReportWorX\RwxService*), and open the AALookupMap.mdb file.
2. Make a copy of the AALookupMapDefault table, give it an appropriate name, and open the new table.
3. Based on the schema of your logging database, edit the information in the AAWX_LookupMap_Actual_Name column. For example, you may want to make these changes:

<i>AAWX_LookupMap_Name</i>	<i>AAWX_LookupMap_Actual_Name</i>
Source	Tag
TimeUTC	EventTimeUTC
ActiveTimeUTC	OriginalEventUTC
ConditionActive	ConditionActive
AAWXLog_TableName	EventLog

NOTE: For AAWXLog_TableName the user can enter multiple names separated by ‘,’ ie Table1;Table2 etc. In this case, the Alarm Analytics Configuration wizards will allow the selection of the base historical alarm table from the list of supplied table names. For the resulting reports to work properly, all used historical alarm tables must have an identical column structure and identical column names.

Not all entries in the AAWX_LookupMap_Actual_Name column need to be filled in, but different analytics use different attributes. Please consult the following table to see some of the attributes that must be mapped in order to use each analytic:

Table 1 - Required Alarm Attributes for Different Analytic Reports

Analytic Report	Required Attribute
Alarm Distribution by Interval	TimeUTC
Alarm Distribution by Interval with Priority	Severity, TimeUTC
Alarm Distribution by Priority	Severity
Alarms from Worst Actors by Interval	Source, Text, TimeUTC
Alarm Rate Distribution by Interval	TimeUTC
Alarm Tag Chattering	ConditionName, Severity, Source, Message, TimeUTC
Alarm Tag Frequency	Source, Message
Average Alarm Rate per Minor and Major Interval	TimeUTC

Analytic Report	Required Attribute
Cross Correlation Analysis	ConditionName, Source, Message, TimeUTC
Operator Changes by Interval	EventType
Operator Response Time	Acked
Standing Alarms by Interval	Acked
Standing Alarms Duration	Acked

Database Expressions

When you do not have a column in your alarm log database that matches one of the attributes in the AWX_LookupMap_Name column, you can use a database expression to calculate a value for that column. For example, if your database does not have an OriginalEventUTC or an ActiveTimeUTC column, but it does have columns for EventTime, EventTimeUTC and Original Event, you can use this expression to calculate the ActiveTimeUTC:

```
DateAdd ("h", DateDiff ("h", EventTime, EventTimeUTC), OriginalEvent)
```

Basically, this expression finds the difference between the EventTime and the EventTimeUTC columns and applies that difference to the OriginalEvent column to get what should be the OriginalEventUTC time. For this example the row in the Lookup Map table would need these values:

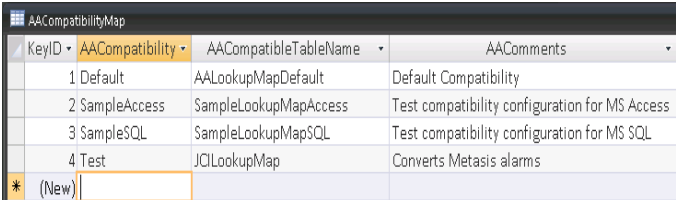
- *KeyID*: 33
- *AWX_LookupMap_Name*: ActiveTimeUTC
- *AWX_LookupMap_Actual_Name*: OriginalEventUTC
- *RequiresCalculation*: checked
- *Expression*: DateAdd ("h", DateDiff ("h", EventTime, EventTimeUTC), OriginalEvent)

NOTE: The exact syntax for these expressions is determined by your logging database. Access and SQL Server databases use slightly different interpretations of the SQL query language, therefore their expressions may be different. For example, the SQL Server expression that is equivalent to the Access expression above would be:

```
DateAdd (hh, DateDiff (hh, EventTime, EventTimeUTC), OriginalEvent)
```

AACompabilityMap Table

Once you have added your new Lookup Map table you will need to add a new entry to the AACompabilityMap table so that Alarm Analytics will know which Lookup Map to use.



KeyID	AACompatibility	AACompatibleTableName	AAComments
1	Default	AALookupMapDefault	Default Compatibility
2	SampleAccess	SampleLookupMapAccess	Test compatibility configuration for MS Access
3	SampleSQL	SampleLookupMapSQL	Test compatibility configuration for MS SQL
4	Test	JCILookupMap	Converts Metasis alarms
(New)			

Figure 4 - AACompabilityMap Table

The AACompabilityMap table includes all named alarm analytics compatibility configurations. For each named configuration, there is a table name of the corresponding historical alarm table column mappings.

To add a new entry, simply create a new record with the next logical KeyID. For AACompabilityName give it a name of your choice. AACompatibleTableName should be the name of the Lookup Map table you created. AAComments is optional and can be whatever comments you would like to add.

Configuring Alarm Analytics Report

Once you have added your translation table, you a re now ready to configure your reports.

1. Create an Alarm Analytic data source as you normally would. If you are not familiar with how to create one, pleaes refer to the application note entitled *Alarm Analytics – Setup Wizards Provide Industry Standard Reports*.
2. Once you have made a connection to the database that contains your historical alarm data, check the Compatibility Mode checkbox and choose the compatibility that you created earlier.

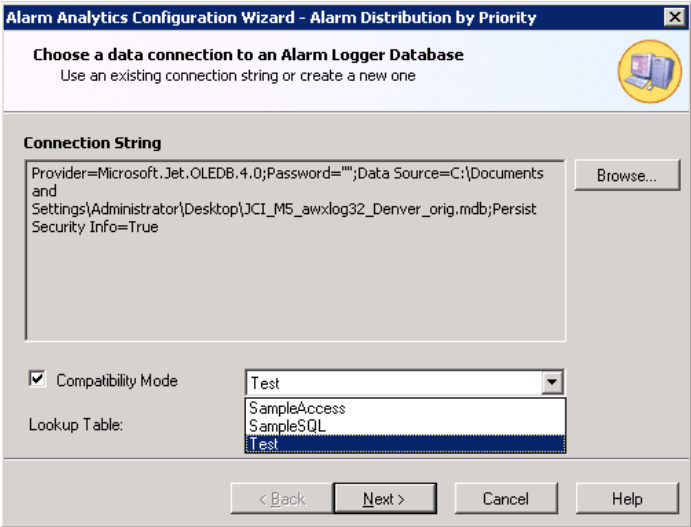


Figure 5 - Alarm Analytics Configuration Wizard: Connection String



Alarm Analytics – Compatibility with Third Party Databases



APPLICATIONS NOTE

October 2013

3. Once you chose the compatibility, it will also tell you the Lookup table associated.
4. Click on Next to continue the rest of the configuration as you would normally.
5. Once you are done configuring the data source and the layout, save the template and close it.
6. Assign the template to a report and execute it.
7. Once the report executes, you can open it to see the data.

Count	Priority	Distribution
0	low	0
1701	medium	0.094206912
12461	high	0.690130704
3894	critical	0.215662384

Figure 6 - Sample Standing Alarms by Interval Data

Summary

To summarize the above wealth of information, the steps required to enable and use the compatibility enhancement feature are as follows:

1. Locate and open the existing AACompatibility.mdb file within the ReportWorX Service Directory.
2. Copy the AALookupMapDefault table, paste and rename it.
3. Modify the AWX_Lookup_Actual_Name column as necessary to map the AWX_LookUp_Name.
4. If necessary, check the RequiresCalculation for custom queries necessary to calculate data for a missing column.
5. Create a new record in the AACompatibilityMap that identifies the newly created Lookup Map table.
6. Configure your Alarm Analytic Report as normal, make sure to enable the Compatibility Mode checkbox and choose the proper translation table.

NOTE: When creating your Alarm Analytics data source in ReportWorX you may see this error: "The specified connection does not point to a valid Alarm Logger database. Please specify another connection." This simply means that there is no AWX_LookupMap table in your alarm log database. To get around this you can copy the AWX_LookupMap table from any AlarmWorX32 Logging database, like the sample located at *C:\Documents and Settings\All Users\ICONICS\Examples32\AwxLog32.mdb* into your alarm log database.