

Description: Guide to alter the AlarmWorX32 Server configuration from your application while a project is running.

OS Requirement: Windows XP / Windows Vista / Windows 7 / Windows 8 / Windows Server 2003 / Windows Server 2008 / Windows Server 2008 R2 / Windows Server 2012

General Requirement: Installation of GENESIS32 9.x.

Introduction

In some applications the alarm server configuration needs to be able to adapt to changes, without having the ability to change the project configuration from the Alarm Server Configurator.

Typical reasons for this include changes that need to be applied by an operator, or eventually automatically.

In the first case, the operator might have few or no knowledge about the AlarmWorX32 Server configuration. There may also be security constraints, which restrict the operator from configuring anything else other than the few elements that fall within his/her scope.

In the second case, changes might need to be applied automatically, for example to react to some operator's actions without his or her knowledge. Changes could also be periodic, having a different or adjusted alarm configuration for the different time of the day.

For example, when running a project to monitor a school, alarms related to fire condition could be more critical during the day when the school is full of children, than at night when nobody is around.

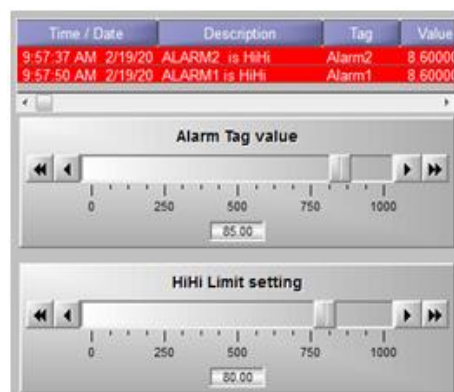
For these or other reasons you may want to alter your alarm server configuration from your application, with or without human interaction, without using the AlarmWorX32 Server Configurator.

Using data sources for alarm parameters

The first approach uses the data sources inside your alarm configuration.

For example, suppose you have used four DataWorX32 registers to store the LoLo, Lo, Hi and HiHi limits for one (or more) of your alarms. While the project is running, you can simply write different values to those registers to alter the configured alarm limits.

Consider the following screenshot, showing a couple of active alarms and two sliders:



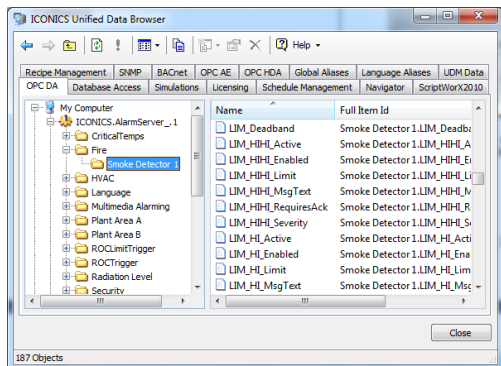


Figure 3 – AlarmWorX DA tags in Unified Data Browser

Direct database manipulation

Another possible approach is to directly modify the configuration database. This is a very low-level approach and should only be carried out if you are well aware of the risks, but still believe you can safely modify your database.

We suggest making a backup copy of the configuration database, when modify it via SQL queries.

To modify the database knowledge about its structure is important, so the first step is to open the database and look for the information that needs to change. As an example, we will alter the HiHi limit of some of the alarms. To do so, the “AWX_Source” table will need to be updated. The “AWX_Source” table has one record for each alarm. Look for the alarm that needs to be modified and browse through the available fields until you find a column named “LIM_HIHI_Limit”. This column will need to be changed, the current value in the field is the current HiHi limit for the alarm.

Modifying that “LIM_HIHI_Limit” field is not the only step required. The Alarm Server also needs to be informed that something has changed. In order to optimize the server’s performances, the server does not periodically reload the full alarm configuration unless it’s really needed. To force it to reload the alarm configuration changes we will also need to tell the server where those have changes occurred. For this purpose the ModifiedSeqNo field is introduced on each record, and it will need to be modified to be the next appropriate value. The next appropriate value can be determined by checking the IDSequences table and taking the number for the AWX_Source row. After this number has been used also ensure to update the IDSequences table with the next appropriate number.

So, for example, to make some changes to the highlighted record (which refers to one specific alarm), the two following fields need to be modified.

ModifiedSeqNo	LIM_RTNTex	LIM_	LIM_I	LIM	LIM	LIM_HIHI_Limit
301	Everything is C	0	1	500		
302		0	1	500		
305	Normal	0	1	500	99	
306	Normal	0	1	500	80	
0		0	1	500		

Figure 4 – Manually changing the configuration database

The next time the server scans for changes, it will see that the ModifiedSeqNo field has changed for this alarm and will reload this particular record and make the changes active in the currently loaded configuration.

The Alarm Server, by default, scans for changes every 20s. If this period is not suitable for your application a different rate can be set in the IcoSetup.ini file. Simply alter the DbPeriod item, which is measured in milliseconds and by default is set to 20000 (for 20s). If a faster scan rate is required, e.g. 5s, set the DbPeriod to 5000 instead.

To recap, to manually change your alarm configuration you will need to prepare the right query to change the fields you need for your application and update the ModifiedSeqNo.

Advanced configuration using BridgeWorX

Whichever approach is preferred, it can be easily combined with the power of BridgeWorX to have your configuration getting automatically updated through the use of BridgeWorX transactions.

Using these tools it is possible to have the alarm configuration change depending on the time of the day, or on any other event that can be used as a BridgeWorX trigger.

Note: You can find additional information on how to create a BridgeWorX transaction to alter your alarm configuration by reading “BridgeWorX - Quick Start” Application Note or BridgeWorX Help file.

Conclusion

We have presented two possible approaches on how to alter you AlarmWorX32 Server configuration from your project. The first approach is definitely more elegant and easier to implement and should be perfect for 99% of the cases.

The second approach might be needed in some very advanced scenario where you want to have full control of the current configuration, being able to alter it at any time, possibly without even having the operator do anything. It is more complicated and risky though.

Combining these approaches with the power of BridgeWorX will let you automatically adapt that configuration to your needs, without having to open the AlarmWorX32 Server Configurator.