AlarmGroup Custom Sounds Code

Play custom sounds for alarm groups.

This document describes how to configure a solution so that it has different AlarmGroups with different sounds. When double-clicking the Sound column, you can choose which kind of windows sound will play when an alarm happens for that specific group.

Solution Example

Alarms Examples

Configuration
- Example 1: Static way
- Example 2: Playing Sound by Scripting

Execution

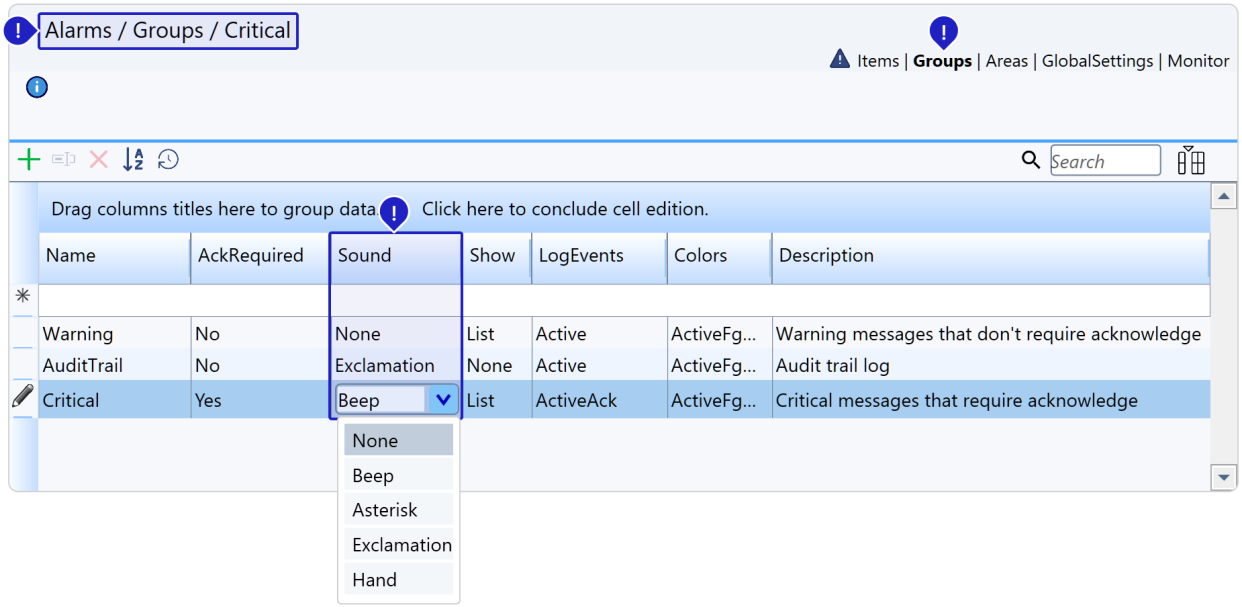
Configuration

It's possible to configure the sounds statically, according to default Windows sounds or dynamically, to advanced users. The following two examples will show how to set both.

Example 1: Static way

To configure a solution so that a group has a different sound when an alarm is active, use one of the following ways:

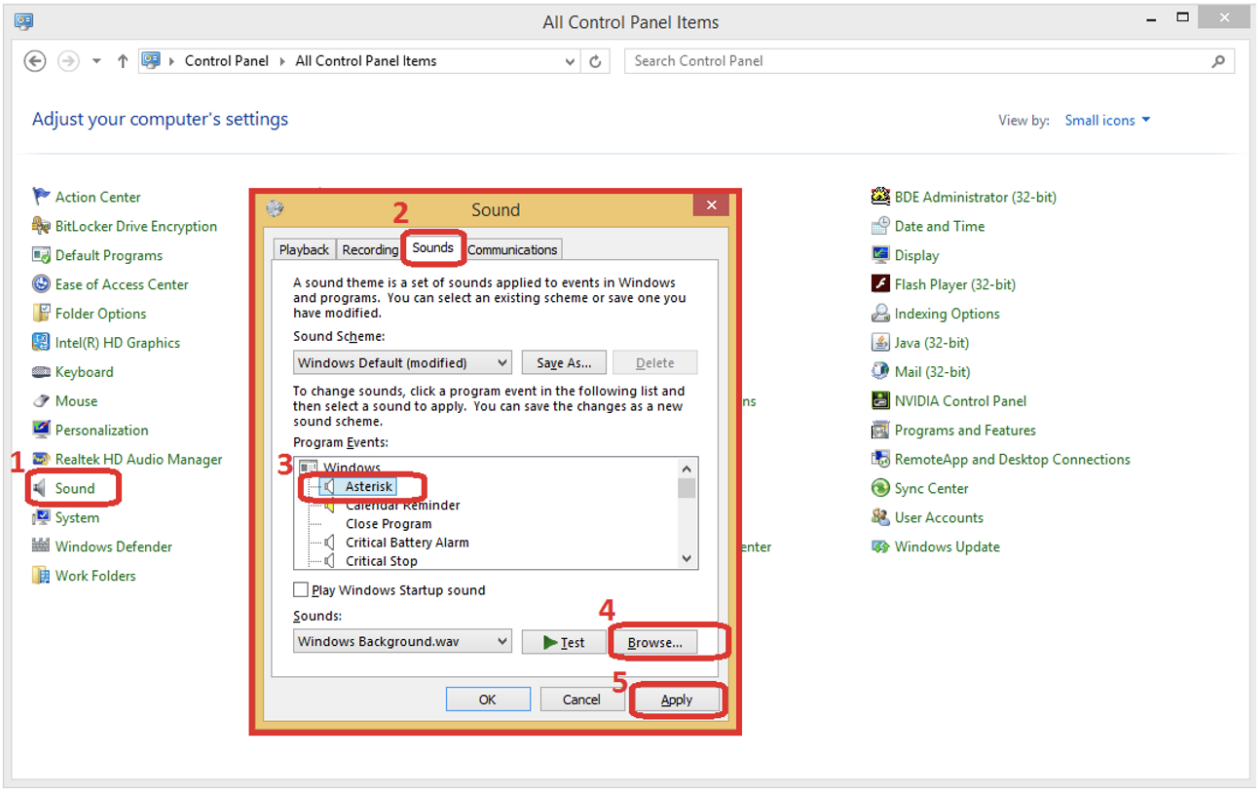
Select one of the default Windows sounds from the Sound column for the specific group.



Use the default Windows sound option as showed above. You can also change the WAV file in the Control Panel on the client computers if you want.

Follow the steps below:

1. Open the Control Panel, and click **Sound** option.
2. In the Sound dialog, click the Sounds tab.
3. Choose the Event sound that you want to change. In this example, we are changing the Asterisk event.
4. Click the **Browse...** button and select a new .wav file for the event.
5. Click the **Apply** button and close the Sound dialog.



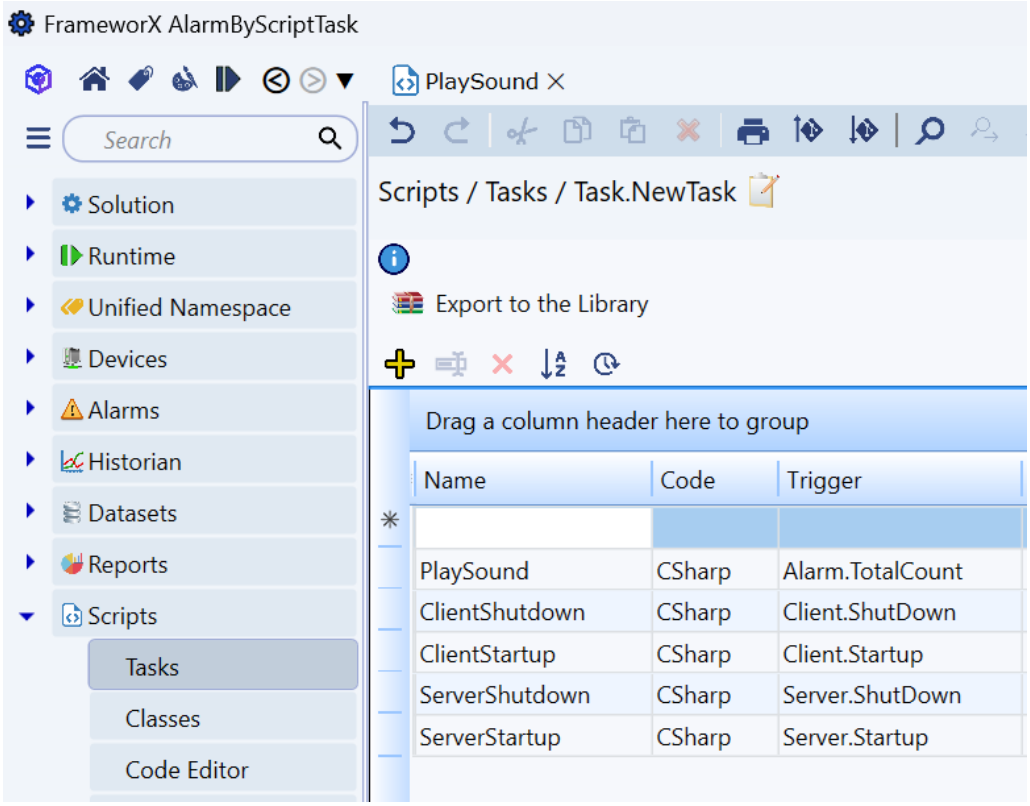
Example 2: Playing Sound by Scripting

If the user needs to play a custom sound when an alarm is triggered, it's possible to do it by scripting. This way, the sound will be played only one time. We support most common extensions, like mp3 and .wav.

> The procedure described here is not set completely in the Alarm Module, but it can be necessary if the user needs to configure any alarm sound into the solution.

In order to play a customized sound signal, set the Sound column on **Alarms / Groups** to be None.

The @Alarm namespace has a property called "TotalCount", it changes its value whenever a new alarm is triggered, so it is possible to run a Task whenever this property changes, registering it as a task trigger.



After that, create a script Task that will run every time the Alarm.TotalCount property changes. This script Task must be a Client Domain.

Below is an example of code you can create to manage the alarm sounds.

```
//Make sure that the script only runs when there is an active alarm
if(@Alarm.TotalCount <= 0)
	return;

//Make sure that the script only runs when the Alarm.TotalCount increases the number of active alarm
if (@Alarm.TotalCount < @Tag.AlarmTotalCountPrevious)
{
	@Tag.AlarmTotalCountPrevious = @Alarm.TotalCount;
	return;
}
@Tag.AlarmTotalCountPrevious = @Alarm.TotalCount;

try
{
	System.Media.SoundPlayer player = new System.Media.SoundPlayer();
	//Get which group that is active has higher priority
	switch (@Alarm.PriorityItem.Group)
	{
		case "Alarm.Group.Warning":
			string filePath = @Info.GetExecutionFolder() + @"\alarm1.Wav";
		break;

		case "Alarm.Group.Critical":
			filePath = @Info.GetExecutionFolder() + @"\alarm2.Wav";
		break;
	}

	//Set the path wav. file
	player.SoundLocation = filePath;

	player.Load();
	player.Play();
}
catch(Exception ex)
{
	@Info.Trace("Error playing audio file: "+ ex.Message);
```

⚠ Since the script Task is a Client Domain, the .wav file needs to be in the correct path that is specified in the code on every client computer.

Execution

Every time an alarm happens when the solution is running, the platform will check which option was configured in the Sound column and play the .wav file configured on the **Control Panel / Sound**. Since the alarm sound is played on the client computer, all the Control Panel configurations must be done on all the client computers.

In this section...
