

# Alarm Send Email

## Installation Notes

This document describe how to configure your project to send an email using the NotificationMethod feature when an alarm is active.

This feature requires FactoryStudio version 2014.2.12 or newer. You can find the NotificationMethod column in the Alarm > Group section.

When you double-click on the NotificationMethod column, it shows every class with the domain Server. After you select the server class, it will show which Method will be used for the call back when an alarm happens. However, the NotificationMethod must follow this prototype:

```
public void NotificationMethod(AlarmEventInfo[] events)
```

---

## Configuration example

In order for your project to send an email when an alarm happens, you need to configure some alarm items. See the image below:



After the items are configured, its time to configure a server class to have a NotificationMethod that receives the alarm events that are being generated. See an example code below:

```

public void AlarmEvents(AlarmEventInfo[] events)
{
    //Protection in case events its null
    if (events == null)
        return;

    //Get the first event
    AlarmEventInfo event = events[0];
    if (event.State != 1)
        return;

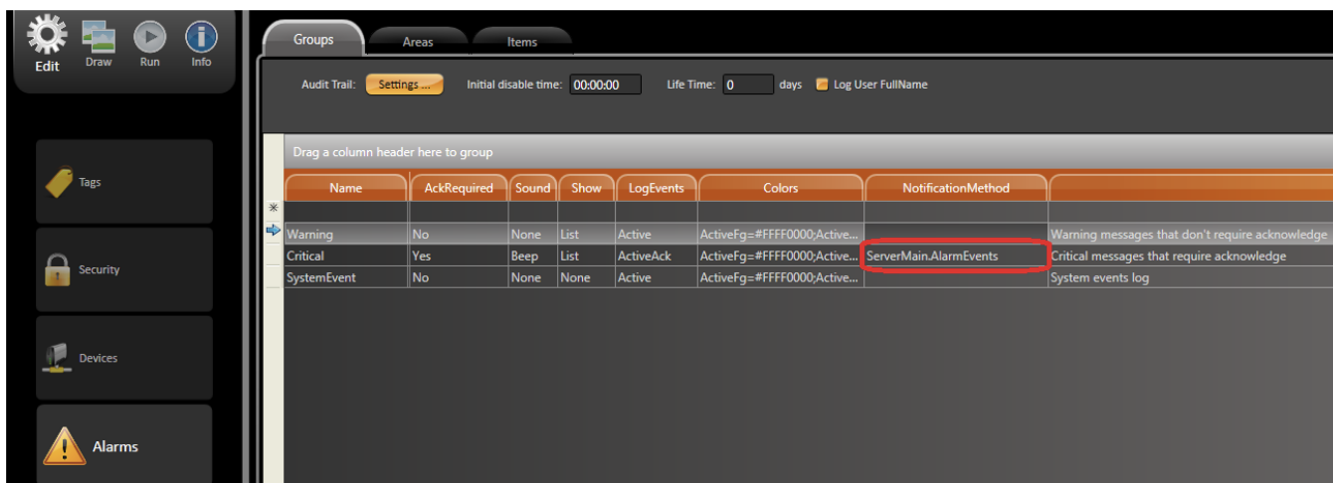
    //Get information about the alarm event to create the body of the email
    string body = "Time: " + event.ActiveLocalTime.ToString() + "\n" +
    "Message = " + event.Message + "\n" +
    "Area = " + event.Area + "\n" +
    "Group = " + event.Group + "\n" +
    "Tag = " + event.TagName + "\n" ;

    //Code to send email
    try
    {
        //Configuring the SMTP Client
        System.Net.Mail.SmtpClient mySmtpClient = new System.Net.Mail.SmtpClient(@Tag.smtpServer, 587);
        mySmtpClient.DeliveryMethod = System.Net.Mail.SmtpDeliveryMethod.Network;
        mySmtpClient.EnableSsl = true;
        mySmtpClient.UseDefaultCredentials = false;
        mySmtpClient.Credentials = new System.Net.NetworkCredential(@Tag.fromEmail, @Tag.
passFromEmail);

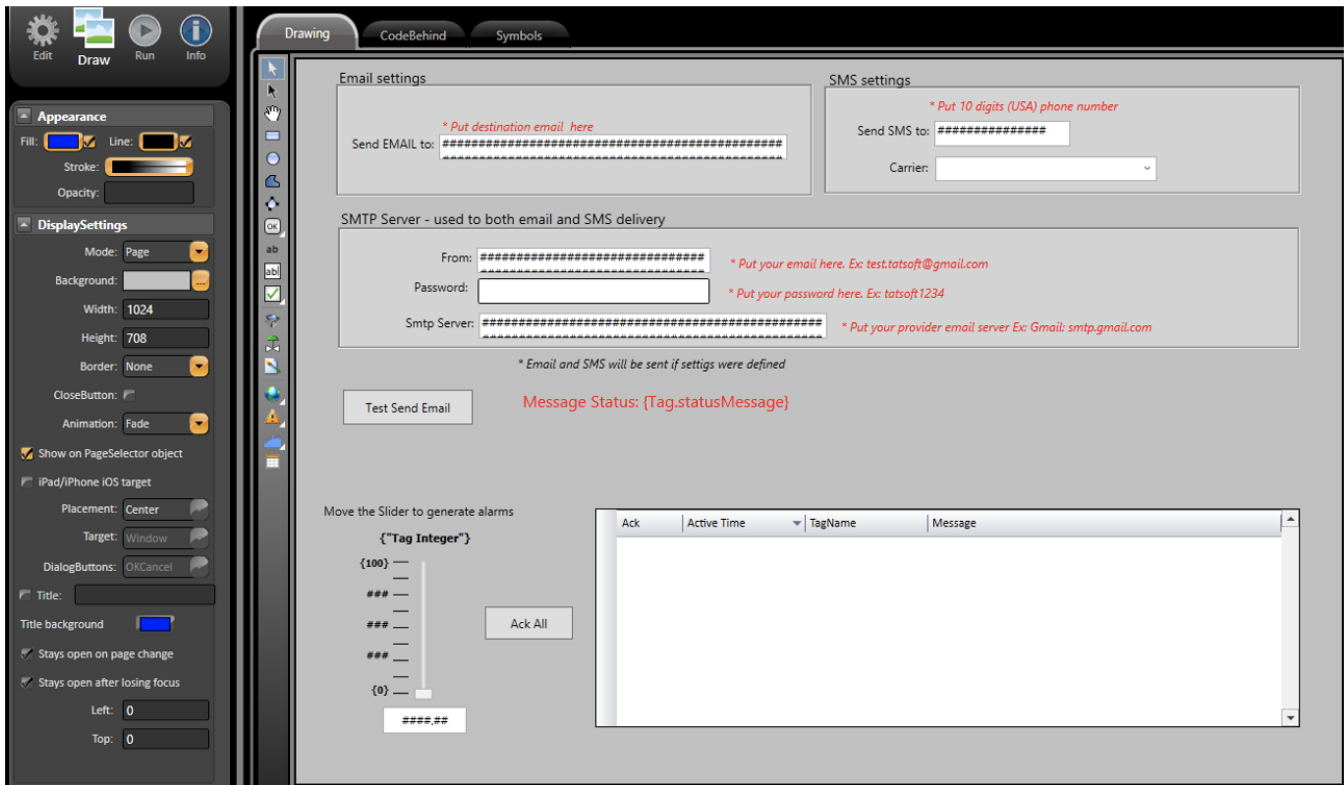
        //Sending the email
        mySmtpClient.Send(@Tag.fromEmail, @Tag.toEmail, "Dmail notification", body);
    }
    catch (Exception ex)
    {
        @Info.Trace("Error sending message: " + ex.Message);
    }
}

```

The next step is to configure the NotificationMethod column in the Alarm>Group section:



The last steps are to create a display, configure the email, and generate some alarms. See an example below:



## Functionality

Every time an alarm happens when the project is running, the method configured in the NotificationMethod column is called. The method will receive an Array of the AlarmEventInfo class as a parameter. The following members can be used for the AlarmEventInfo class.

```
DateTimeOffset AckTime
    DateTimeOffset ActiveLocalTime
    DateTimeOffset ActiveTime
    string Area
    string Category
    string ColorBG
    string ColorFG
    string Comment
    int Condition
    DateTimeOffset DateCreated
    string Deadband
    string DetailsValue
    int Disable
    TimeSpan Duration
    string Group
    string Level
    string Limit
    string Limit1
    string Limit2
    string Message
    DateTimeOffset NormTime
    int Priority
    int Quality
    string Setpoint
    string SetpointDeadband
    int State
    string TagName
    int UnAck
    string UserName
    string Value
```

Once in the NotificationMethod, you can do everything you want; send an email, send a message box, speech an alarm message, etc.