

NotifyMe (2.0) - Programming

05/13/2024 6:32 pm EDT

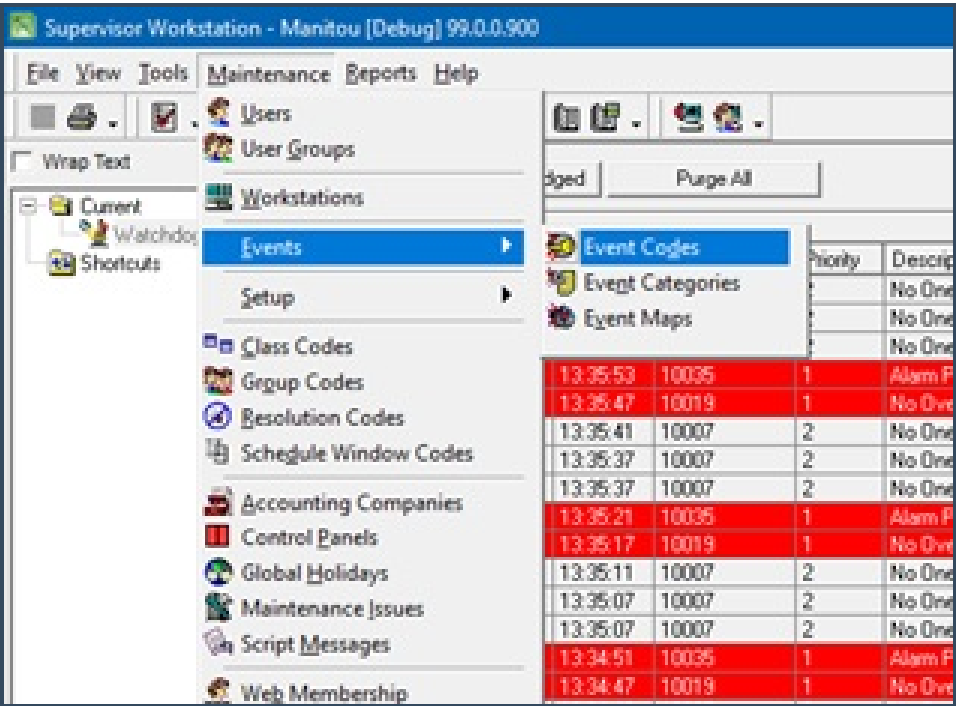
Overview

The intention is for the vast majority of programming for NotifyMe to be done at the Event Code level. This allows it to be set up once and managed/modified in one central place. However, there does need to be some way to provide individual control at lower levels so that specific situations can be addressed. This is provided by overriding the command with a different command or perhaps no programming at all. Both of these are shown in the following sections.

NotifyMe is not intended to be applied globally to every event code. As such, using a "*" TX Type Programming does NOT work. NotifyMe will not perform the expected functions if you attempt to use this type of programming. Instead, the programming should be applied at each Event Code you wish to use individually.

NotifyMe Event Code Programming

For NotifyMe to function, a few programming commands must be assigned to each event code type. This tells Manitou what NotifyMe functionality to allow, how long the alarm is held in a pending state, and what happens when the NotifyMe user responds. Event Code programming can be found in SWS under *Maintenance > Events > Event Codes*



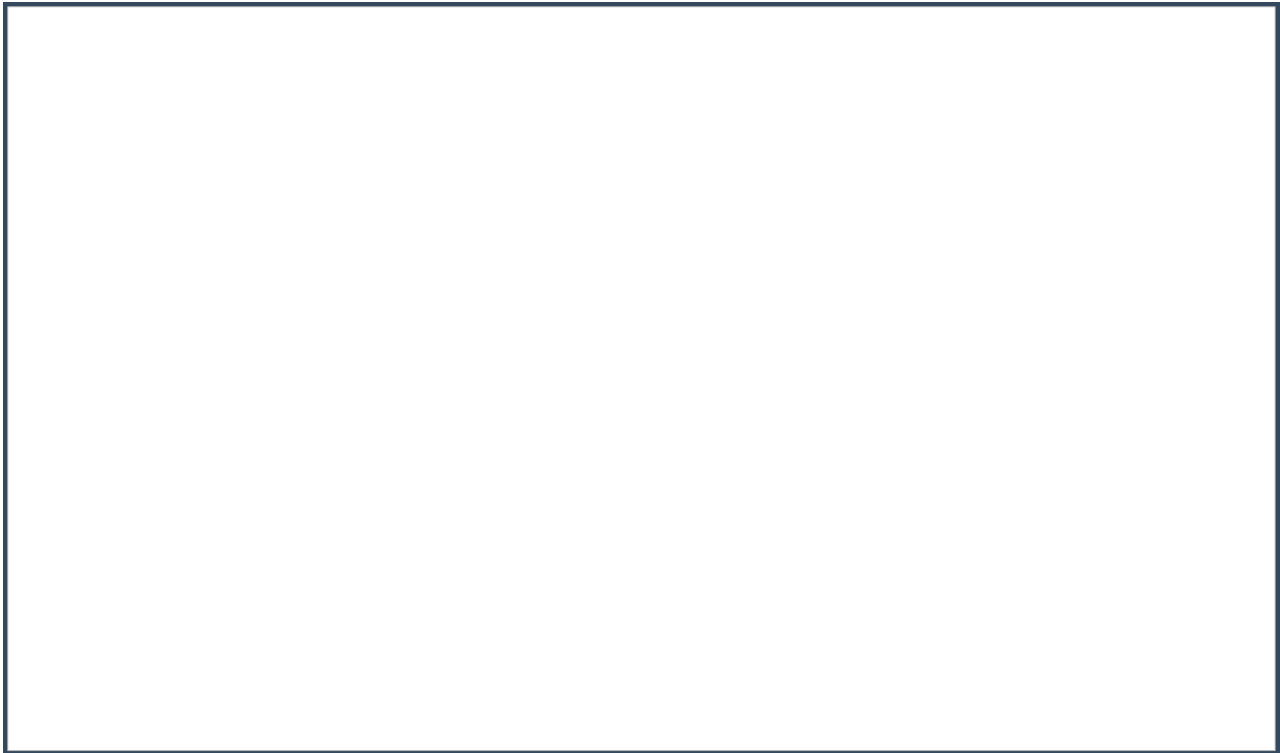
Each event code that you want to use with NotifyMe must be updated to add the CanCancel() and Cancel() programming commands. This is done by selecting the event code, editing it, and using the wizard to both of these commands to the TX Programming section. There must also be a reset command if that functionality is to be implemented. This means that to get complete functionality, multiple Event Codes must be configured.

These must match what is set in the Integration Gateway Config for a given evcode name (verify, reset, restore, and verified)



CanCancel() command

Here you must set the Timeout desired (in seconds) where the signal will be held in a pending state before dropping into the alarm queue. Choose the Event to match the Event Code that you are applying this programming on and also select if the NotifyMe user will be allowed to Abort the signal once they have received the notification. This means the alarm can be closed before ever getting to the alarm queue. Care must be taken to choose settings that make sense for the alarm in question.



You MUST have both the CanCancel and the Confirmed on an event code to allow the NotifyMe Verify functionality to work. The CanCancel will set the time that it remains hidden from the Alarm Queue and the Confirmed allows it to be verified by the NotifyMe user. These must be added together in one programming command. The timeout value for both must match!

ResetTimeout() Command

This is used to reset the timer that keeps the alarm from going into the Alarm Queue to be handled by an operator when the NotifyMe user presses the "More Time" button in the App. Typically this is set on the RN event code but must match what is set for the reset value

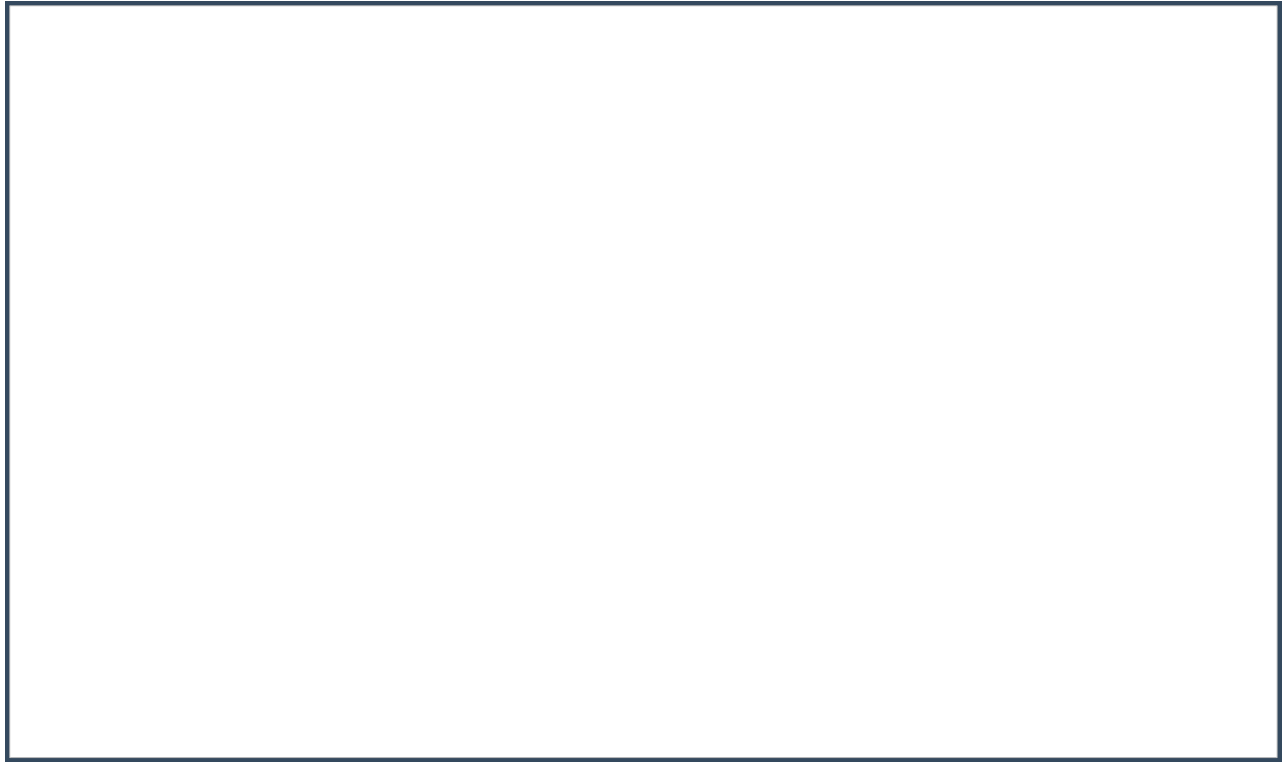
in the Integration Configuration.



Because the `ResetTimeout()` command requires an event, there must be multiple of these created and programmed for each event code that you wish this to apply to. This is one of the major setup inefficiencies that is corrected in the second gen of NotifyMe on the Manitou side.

Cancel() Command

This is what allows the signal to be cancelled by the NotifyMe user. This must be set on the event code listed in the Integration Configuration for restore.



Transmitter Type Level Override

In order to provide customization for specific use cases, these programming commands can be overridden at several levels. The first is at the Transmitter Type. Here specific programming can be applied to any account that uses this Transmitter type. It can be used for separating particular dealer/branches or just grouping similar accounts to use common settings.



This can also be found in OWS under *Maintenance > Transmitter Types*.

Once there, select the transmitter type you want to apply custom programming to and switch to the Programming Tab. Under TX Programming, you would add in the programming commands that you want to use for just this transmitter type as shown in the above section. This would be done for each event code that you want to override at this level.

To override and NOT have NotifyMe used at all for a given event code just for this specific transmitter type, you must add an Event Code programming line WITHOUT a command at all.



Overriding effects ALL programming for this event code not just NotifyMe. If there was other programming and you remove programming altogether, that other programming will not happen. If the other programming is needed it must be added back at this level in order to function.

Account Level Override

The final place that Event Code programming can be applied is at the account level. Here individual accounts can have custom programming defined for each system. This would allow very specific accounts to have a modified version of the NotifyMe programming on a per-case basis. This can be found under each system in the Programming section. Selecting “All” in the Show Merged TX Programming for TX section will allow you to see if any Transmitter Type overrides have been applied. This will display for each of the Transmitters under that system.

This will NOT show the Event Code programming. You must look/know that NotifyMe programming has been applied at that level.

If there is no programming displayed for the event code you wish to override, then following the same edit/add process as shown in the Transmitter Type Level Override section can be used. In ManitouNeo, if there is programming already applied at the Transmitter Type Level, you can simply click on the line for that transmitter type and the system will ask if you wish to customize the programming.

The line will turn white indicating that the programming has been copied to the account level and can be modified as needed. Then you would edit the programming to make the changes as you need for this specific account.

Overriding effects ALL programming for this event code not just NotifyMe. If there was other programming and you remove programming altogether, that other programming will not happen. If the other programming is needed it must be added back at this level in order to function.

In OWS, you just simply modify the shown Transmitter Type command and when you save you will see both the old and new lines. The lines shown in white are the account level ones and they are what will take precedence over the TX Type programming.

Conclusion

With these three levels of programming possible, we believe that NotifyMe can be set up very quickly for the majority of a central stations needs by applying the bulk of the programming globally at the Event Code level. There will always be a need for adjusting some of this programming for specific conditions but this will be a lot less effort that attempting to apply programming on every account. In addition, we have greatly simplified the NotifyMe programming into a single command with a wizard.

