NotifyMe (2.1) Programming

05/13/2024 6:08 pm EDT

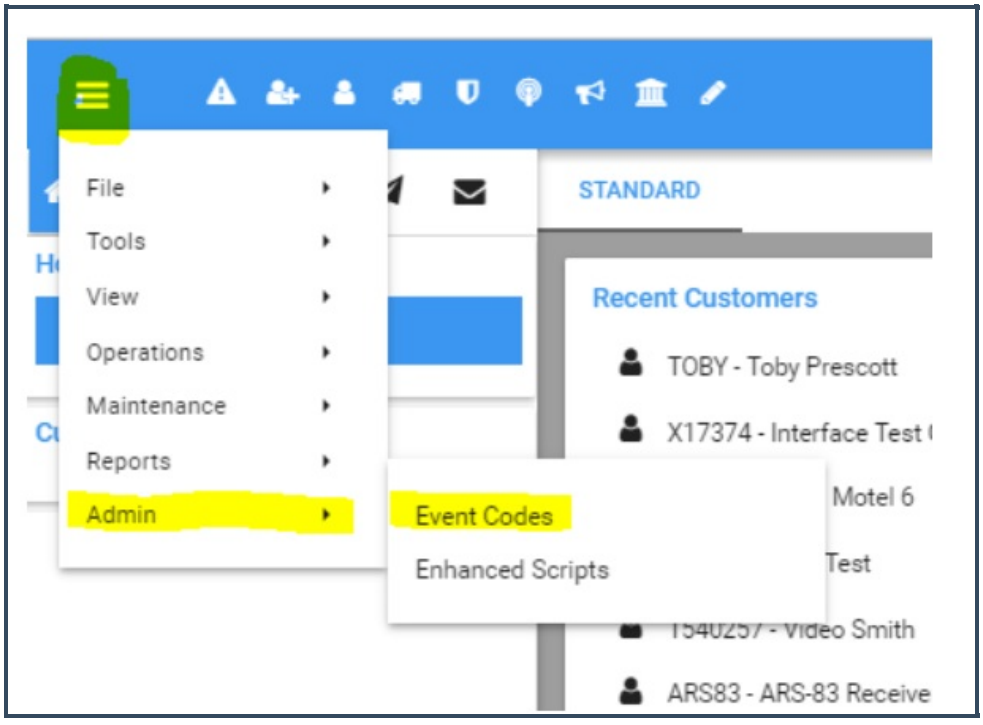
Overview

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 needs 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.

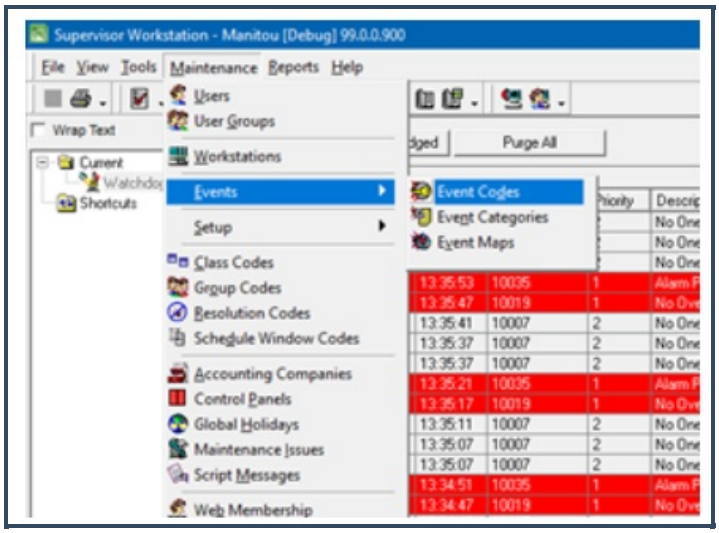
NotifyMe is not intended to be applied globally to every event code. As such, using a "*" TX Type Programming does NOT work. NotifyMe will not perform the expected functions if you attempt to use this type of programming. Instead, the programming should be applied at each Event Code you wish to use individually.

NotifyMe Event Code Programming

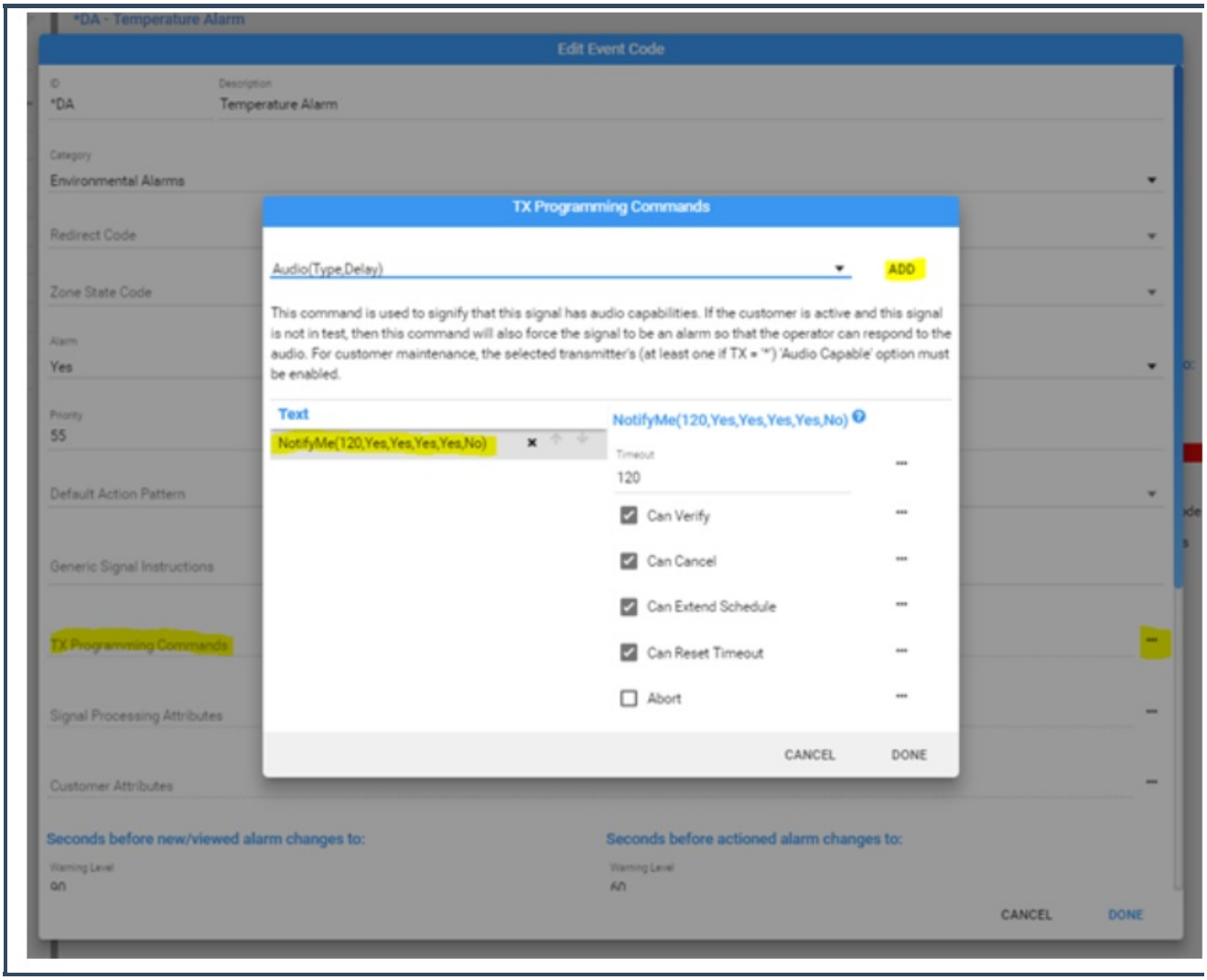
For NotifyMe to function, a programming command must be assigned to each event code type. This tells Manitou what NotifyMe functionality to allow, how long the alarm is held in a pending state, and what happens when the NotifyMe user responds. To apply programming at the event code level, we have added Event Codes to the Admin section in ManitouNeo.

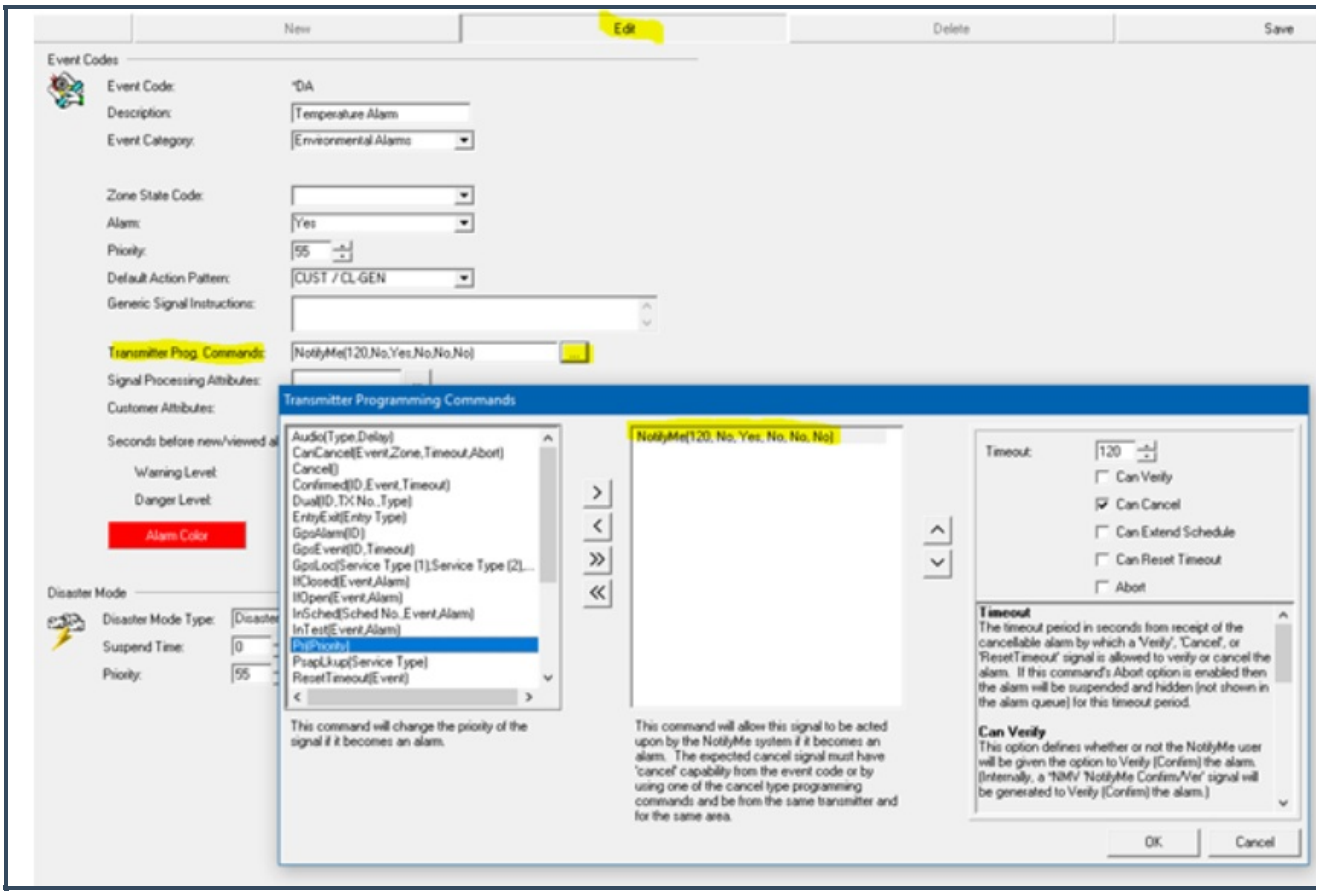


It can also be found in SWS under Maintenance > Events > Event Codes



Each event code that you want to use with NotifyMe must be updated to add the NotifyMe programming command. This is done by selecting the event code, editing it, and using the wizard to add the NotifyMe command to the TX Programming section.

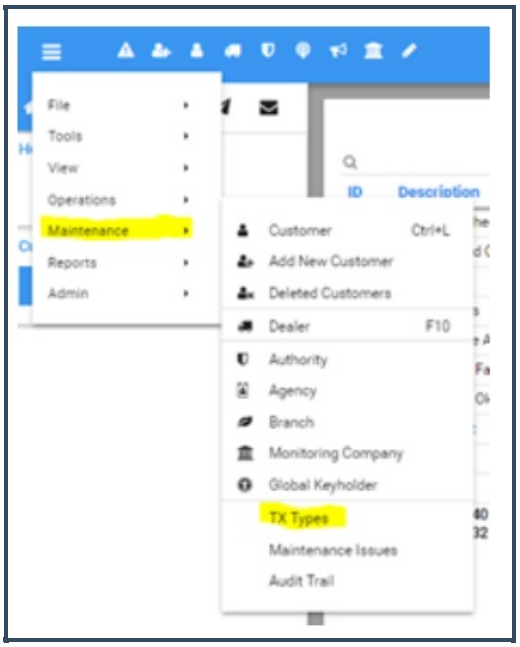




Here you must set the Timeout desired (in seconds) where the signal will be held in a pending state before dropping into the alarm queue. Also select if the NotifyMe use will be allowed to Verify, Cancel, Extend, Reset, or Abort the signal once they have received the notification. Setting these will cause the corresponding controls in the app to display or be hidden. Care must be taken to choose settings that make sense for the alarm in question.

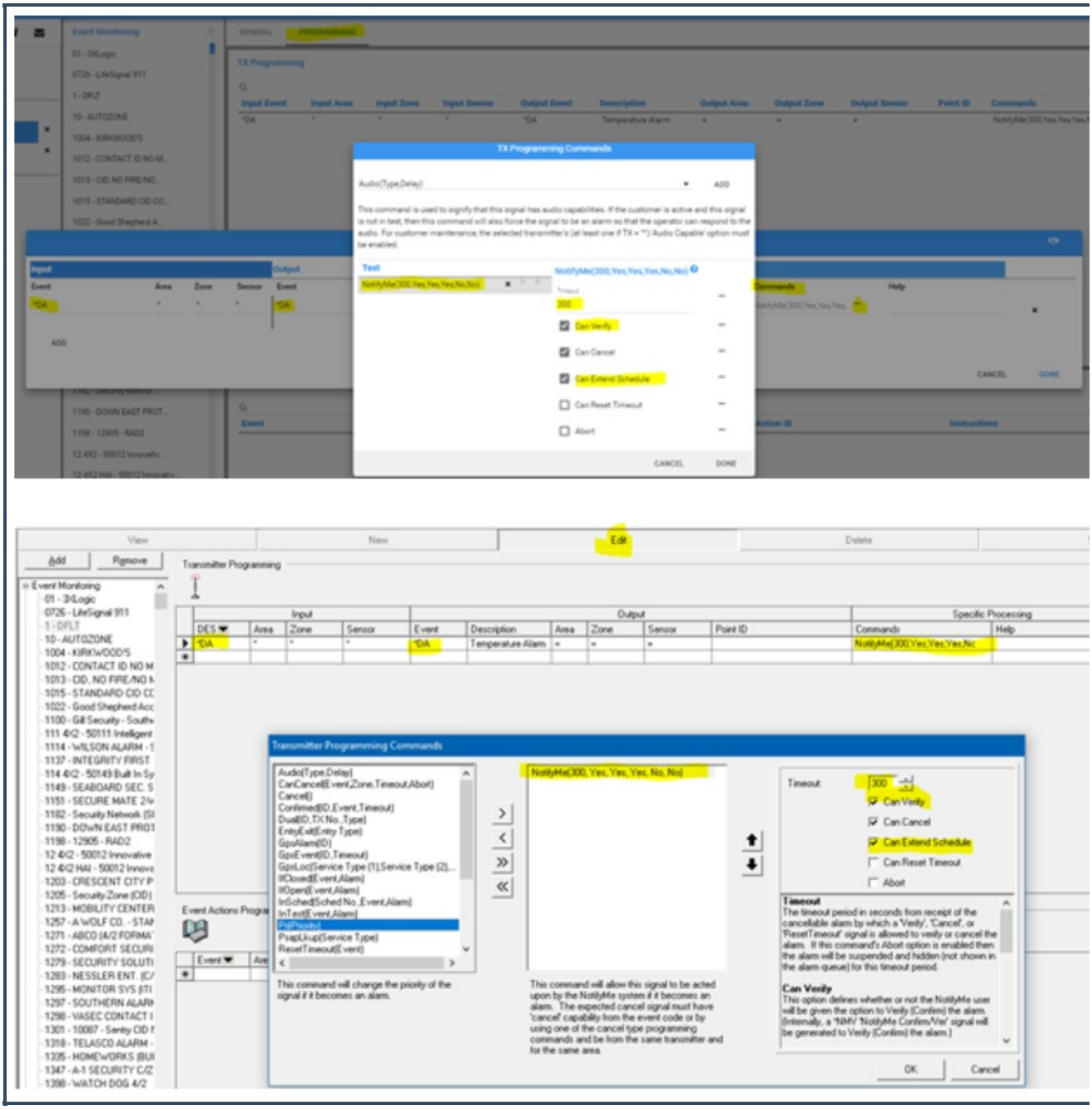
Transmitter Type Level Override

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 modified NotifyMe programming command that you want to use for just this transmitter type. This would be done for each event code that you want to override at this level.



To override and NOT have NotifyMe 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.

IMPORTANT: Overriding affects ALL programming for this event code not just NotifyMe. If there was another programming and you remove the programming altogether, that other programming will not happen. If the other

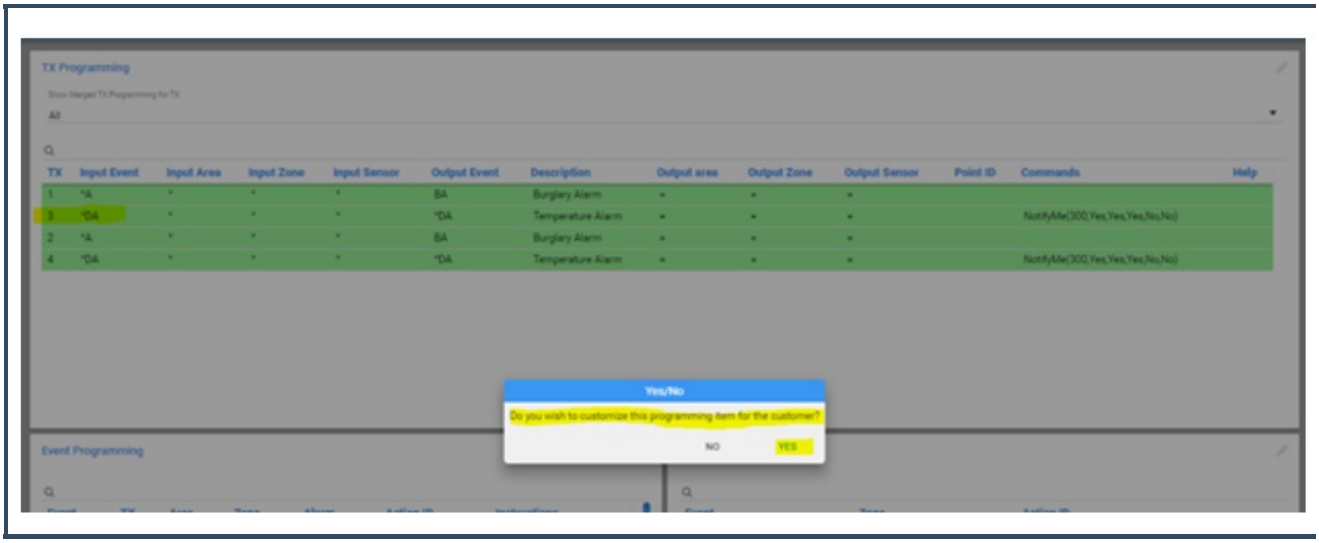
programming is needed it must be added back at this level in order to function.

Account Level Override

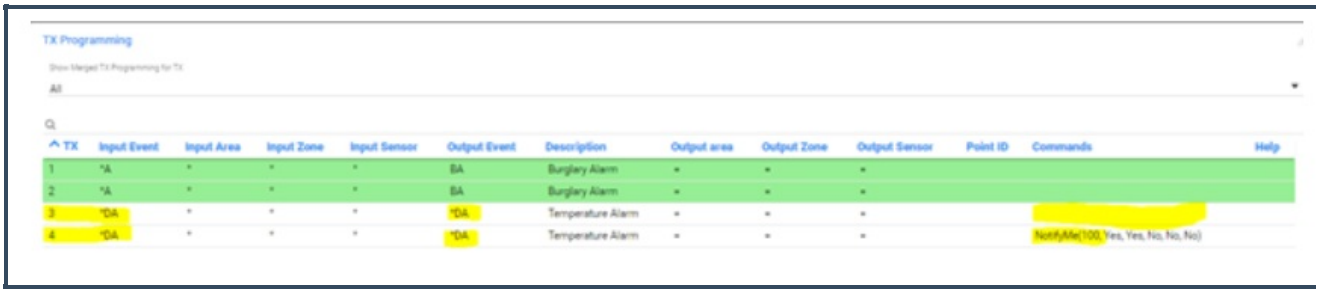
The final place that Event Code programming can be applied is at the account level. Here individual accounts can have custom programming defined for each system. This would allow very specific accounts to have a modified version of the NotifyMe programming on a per-case basis. This can be found under 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.

NOTE: 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.

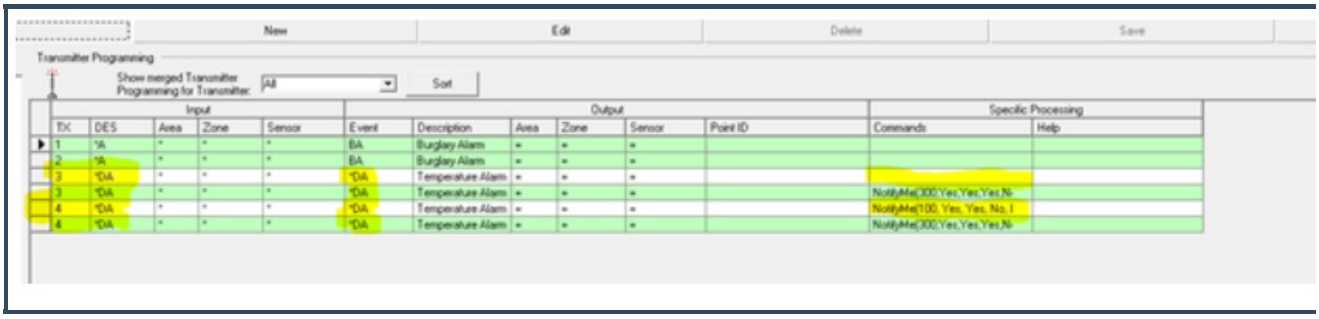


The line will turn white indicating that the programming has been copied to the account level and can be modified as needed. Then you would edit the programming to make the changes you need for this specific account.



IMPORTANT: Overriding affects ALL programming for this event code not just NotifyMe. If there was another programming and you remove the 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, 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 central stations need 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 than attempting to apply programming on every account. In addition, we have greatly simplified NotifyMe programming into a single command with a wizard.