

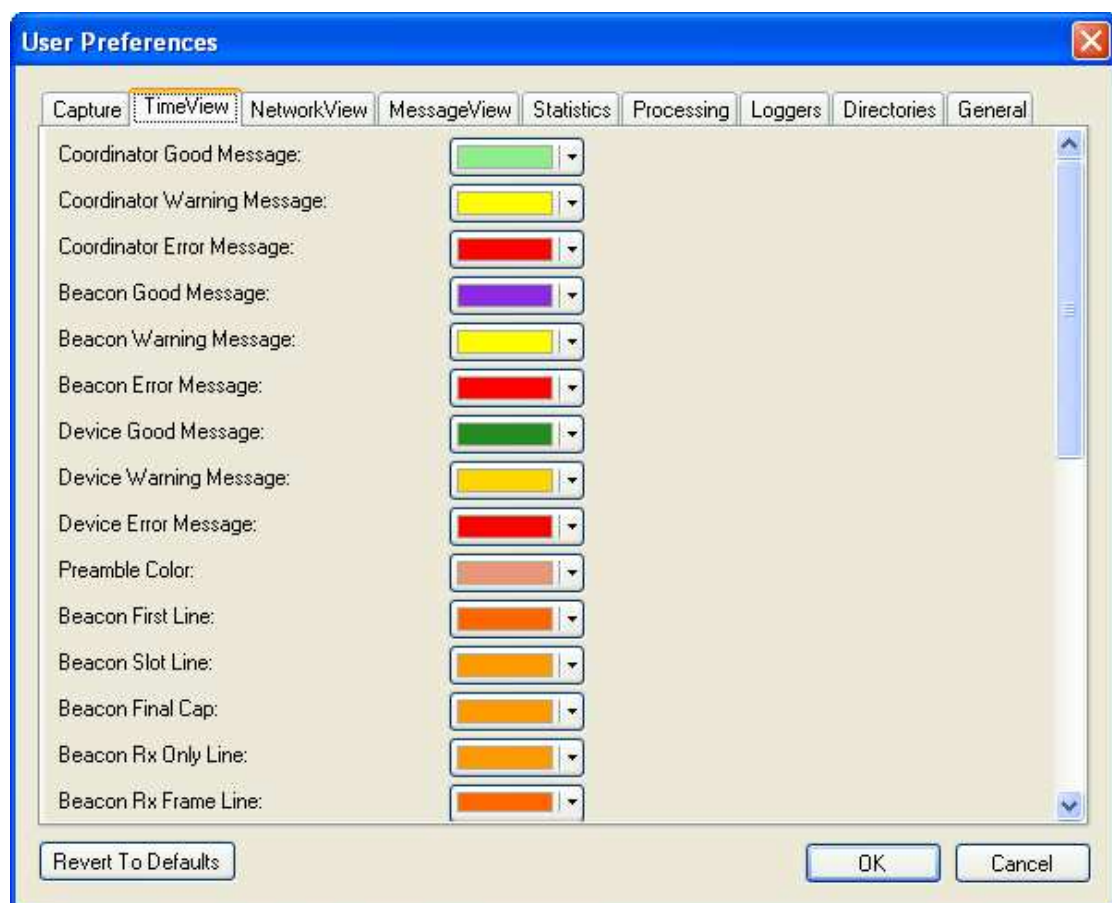
Perytons™ Flexibility

The Perytons™ Protocol Analyzers provide a variety of flexibility levels for the user. Starting with the basic look and feel preferences available to all users, and all through to a full environment allowing the user to build his own protocol into the analyzer, create charts and more.

Preferences

Available in all basic Perytons™ Protocol Analyzer models.

Allows the user to set the colors of different message types, connections (colors and line thickness), etc.



Using Open Source Rules written by Perytons™ or by Peryton-Monitoring Add-On holders

Available in all basic Perytons™ Protocol Analyzer models.

Any Perytons™ Protocol Analyzer user can use Open Source Rules written for the analyzer.

Such an Open Source Rule can generate charts, filter messages in Time View and Message View windows, change the device icon in the Network View, generate events and alarms, and

more (for further information regarding Perytons™ Open Source Rules see the Perytons™ Open Source Rules whitepaper).

Write your own Open Source Rules

Available as part of the Perytons-Monitoring Add-On license.

Allows to write/modify Open Source Rules and sign them to make them available for other Perytons™ Protocol Analyzer users. The Open Source Rules are written using C# (C-sharp) convention and no external development environment is needed for writing or debugging them. Signed Open Source Rules written can be shared with other Perytons™ Protocol Analyzer users regardless of their product license model.

Other than the basic Open Source Rule features, the Open Source Rule has access to PC resources allowing the user to create sophisticated code that can send messages over LAN according to defined scenarios, send information over the internet, etc.

SDK – Software Development Kit

Available as part of the Peryton-SDK Add-On license.

Allows writing a complete protocol into the Perytons™ Protocol Analyzer as well as adding protocol layers on top of layers already supported by the Perytons™ Protocol Analyzer. The development of the new protocol is easy and based on predefined templates to a variety of fields, yet fully flexible due to the use of C# coding, and the ability to override any of the template's features.

Using the SDK requires Microsoft Visual Studio development environment.