Coding Exercise

At Navico all code that is submitted to our source code repository must first be peer reviewed. This ensures that the developer is following our coding standards, and ensures that a basic level of quality has been met.

To complete the exercise:

Attached are brief code samples, please review and submit the following:

1) **A list of deficiencies in the existing code, and supporting reasons for why you think the code should change**

   This can be brief, provided you clearly state why the code should change.
   You can provide supporting links if you think this is appropriate. For example:

   a. *Class tFoo is a Singleton. Singletons are considered an anti-pattern. (See singletons are pathological liars for more info)*
   b. *Variable m_Bar is declared type float, but is cast to an int and used as an int. Re-write so that m_Bar is an int to avoid an unnecessary cast*

2) **The code sample rewritten to the best of your knowledge**

   New code should support C++11 features. You can modify the code in any way you like and you can assume that changing the API of tChartPanel (in anyway) is not an issue.
   For example, you should feel confident to change the parameters of a method (if it makes sense to do so) without worrying about breaking the calling code.

   Some additional information that is relevant to the sample code:

   a) *Assume a tBigObject is a large object and expensive object to create/copy*
   b) *tWarningDialog and tErrorDialog are modal dialogs. They should never be created on the stack as there is a subtle bug that can lead to a crash. These dialogs should always be created on the heap.*
   c) *Assume that a tBorder has a method float getBorderWidth()*
   d) *You can use the STL and boost libraries (or equivalent) if necessary*
   e) *You are encouraged and recommended to produce modern C++ code (at least C++11)*

Some resources that may help you to complete the exercise:

a) Scott Meyers Effective C++ series
b) Resource Allocation is Initialisation (RAII)
c) SOLID Programming Practices
d) STUPID Programming Practices

The code sample is to display a chart widget inside a panel (a UI container). However, *what* the code is doing is not important, what is more interesting for this exercise is *how* the code is trying to do it.