Unit Testing Best Practices for Qt-Based Classes

Certain semantic conventions used in Qt-based applications dictate specific ways in which unit tests should and should not be constructed. In particular:

1. Many Qt classes take the Parent pointer as a constructor parameter. Not all subclasses of the base type specified for that parameter are semantically valid for any given Qt class.
2. Qt Parent objects take ownership of their children, including memory management. This implies that Parents delete children when they are deleted themselves.

Implications for automatically-generated test cases are as follows:

- Automatically-generated tests create primary and secondary test objects on the stack. In most cases, this will conflict with convention #2 above, resulting in memory errors during test case execution, with stack traces most likely pointing to the closing brace of a destructor.
- In automatically-generated tests, C++test will attempt to use any available subclass type for an argument of a base type. In many cases, this will conflict with issue #1 above.
- If tests require definition of the QApplication variable, the macro definition `- DCPPTEST_INIT_QT=1` should be added to the compiler options. To use the Qt console application,also add the macro `-DCPPTEST_INIT_QT_CONSOLE=1`.

We recommend the following procedure for unit testing Qt-based classes:

1. Automatically generate unit tests; this will give you a template.
2. Keep one of the generated tests as a template, preferably one that passes (does not throw an exception) when run.
3. Using the one test as a template, replace test object allocations by declaration with those on the heap (using new).
4. At the end of the test, if a parent object is used, delete only that—whether it is the primary test object or a secondary test object.
5. Validate the one test by using a debugger to observe the final test object states and inserting appropriate assertion macros.
6. Add more test cases (using the GUI dialog option, and using the first validated test case as the template).