

DOxygen Style Guide CS1C Project

For detailed descriptions of classes and functions-we should use option 2 (QT option) plus it matches the style of the brief description:

There are several ways to mark a comment block as a detailed description:

1. You can use the Javadoc style, which consist of a C-style comment block starting with two *'s, like this:

```
/**
 * ... text ...
 */
```

2. or you can use the Qt style and add an exclamation mark (!) after the opening of a C-style comment block, as shown in this example:

```
/*!
 * ... text ...
 */
```

In both cases the intermediate *'s are optional, so

```
/*!
... text ...
*/
```

is also valid.

For brief descriptions of classes and functions:

```
//! Brief description.
```

For variables:

If you want to document the members of a file, struct, union, class, or enum, it is sometimes desired to place the documentation block after the member instead of before. For this purpose you have to put an additional < marker in the comment block. Note that this also works for the parameters of a function.

Here are some examples:

```
int var; /*!< Detailed description after the member */
```

Example:

Examples

Here is an example of a documented piece of C++ code using the Qt style:

```
///  
/*!  
    A more elaborate class description.  
*/  
class QTeststyle_Test  
{  
public:  
  
    ///  
    /*! More detailed enum description. */  
    enum TEnum {
```

Generated by Doxygen 1.8.15

4.1 Special comment blocks

21

```
        TVal1, /*!< Enum value TVal1. */  
        TVal2, /*!< Enum value TVal2. */  
        TVal3 /*!< Enum value TVal3. */  
    }  
    ///  
    /*! Enum pointer.  
    /*! Details. */  
    *enumPtr,  
    ///  
    /*! Enum variable.  
    /*! Details. */  
    enumVar;  
  
    ///  
    /*! A constructor.  
    /*!  
    A more elaborate description of the constructor.  
    */  
    QTeststyle_Test();  
  
    ///  
    /*! A destructor.  
    /*!  
    A more elaborate description of the destructor.  
    */  
    QTeststyle_Test();  
  
    ///  
    /*! A normal member taking two arguments and returning an integer value.  
    /*!  
    \param a an integer argument.  
    \param s a constant character pointer.  
    \return The test results  
    \sa QTeststyle_Test(), QTeststyle_Test(), testMeToo() and publicVar()  
    */  
    int testMe(int a, const char *s);  
  
    ///  
    /*! A pure virtual member.  
    /*!  
    \sa testMe()  
    \param c1 the first argument.  
    \param c2 the second argument.  
    */  
    virtual void testMeToo(char c1, char c2) = 0;  
  
    ///  
    /*! A public variable.  
    /*!  
    Details.  
    */  
    int publicVar;  
  
    ///  
    /*! A function variable.  
    /*!  
    Details.  
    */  
    int (*handler)(int a, int b);  
};
```