This document contains steps associated with some procedures for modifying the omc code base to extend functionality.

**Add a new symbol to the Symbol hierarchy**

The following steps were documented during the addition of the new symbol DimensionSymbol to the Symbol hierarchy.

1. In the ompp solution, set the active project to omc.
2. Choose an existing symbol with similar properties, e.g. TableExpressionSymbol has some similarities to the new DimensionSymbol.
3. Create copies of the declaration (.h) and implementation files (.cpp) of the existing symbol, and rename them using the new symbol.
4. Use “Add” in the Tortoise SVN context menu to mark the two new source files (DimensionSymbol.h, DimensionSymbol.cpp) for subsequent upload to SVN (but don’t check in yet).
5. Use Solution Explorer to add the new header file to the Header Files group, and the new implementation file to the Source Files group.
6. Open the file ast.h, and add a #include statement for the new header file. Close ast.h.
7. Open the two new files for editing, and close any other files open in the editor.
8. Use global replace (Ctrl-H) on the two open files to change all uses of the original class name in the copied files (TableExpressionSymbol) to the new class name (DimensionSymbol). Global replace settings are case sensitive, whole word, all open documents.
9. At the beginning of the class declaration, check and change the base class if necessary, and verify that the typedef declaration for ‘super’ 3 lines below corresponds. Very important, as this is the mechanism used to implement hierarchical calling chains. Bugs here have odd symptoms and can be difficult to track down.
10. Edit doxygen comment for the new class in the header file.
11. Continue with minimal implementation related to the new .h and .cpp files, but  
    - avoid references to the new class in other code for the moment (e.g. grammar file parser.y)  
    - avoid implementing new class-specific functionality  
    Typical changes at this point include:  
    - remove irrelevant members  
    - remove irrelevant arguments for constructor  
    - remove #include for unused header files  
    - verify and edit doxygen comments for all class members in the .h file.
12. Do trial compilation of the implementation file (TableDimensionSymbol.cpp), using Ctrl-F7, and fix errors as required.
13. Build omc. There should be no errors.
14. Create code snapshot using SVN or other method (optional).
15. Continue with implementation of functionality for the new class.