

TreeTableView Implementation Details

It is not my intention to discuss the implementation in a huge depth, but I do think it is useful to clarify that the intention in building out the TreeTableView implementation is to reuse as much of the existing implementation code from TableView and TreeView wherever possible, without having to resort to copy/paste. Therefore, whilst it is very critical that the [TreeTableView API](#) be able to stand on its own, I feel I should clarify that many of the decisions made (particularly around extracting common *Base classes) was driven by a desire to minimise code duplication in the implementation. I believe that the current level of extraction has taken us as far as we can go in reducing code duplication, and that there is in fact now very little TreeTableView specific code in the implementation - most classes in the implementation related to TreeTableView are simply extending an abstract class and providing the necessary code (normally 5-10 methods), most of which are one-line methods returning some TreeTableView-specific implementation of a public API.

In short, if we want to swing back the other way and reduce the amount of abstraction (or the number of newly-introduced classes in the public API), we will need to increase the amount of code that is possibly duplicated in the implementation. It's a fine balancing act :-)