I'm struggling to understand why you're making it so complicated.

-If you are automatically unregistering Dispatchables on destruction you have no need to store Dispatchables in Ptr<>s and so can elliminate the overly complex PTEABLE branch.

-If you make DispatcherGen use Dispatcher objects internally (as I did) or even derive it from Dispatcher you can eliminate all of the complex Any stuff with sources. Keep it simple: Dispatcheables should only be registerable with Dispatchers and derived classes. Use the advantages of polymorphism.

-You don't need the source keys or the source key cache. If no key is specified by the caller to Unregister you can just iterate though the dests values looking for the correct pointer. This is potentially slightly slower, but if the developer is worried about performance then they should be using keys anyway.

-As I said before, do away with the NOMAP branch, the performance and memory usage gains are negligeable for the added code complexity. Having #defined branches through the code is really horrible, if you absolutely must have the branches then separate them completly or make them templated or something, currently it's extremely hard to read and understand the code.

Sorry, I don't want to seem too critical and my solution isn't the only way, but the whole thing just seems much more complicated (and thus diffcult to maintain and use) than it needs to be. Are you imagining some sort of use-case that I haven't considered?