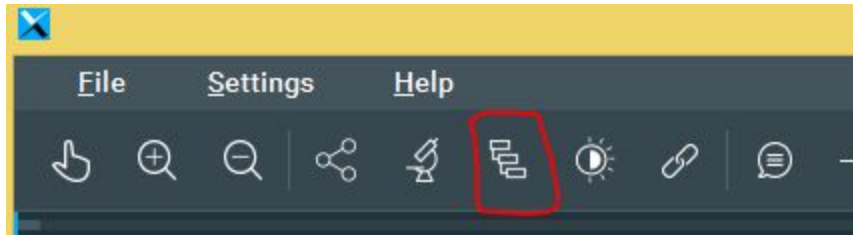


Getting started with API

Before we continue, please note that since API is still in development it might contain bugs so please do not use it on a real account (even though there is some level of protection that should try to prevent it from happening it's not guaranteed to work).

First, please open some file in replay and make sure you see this button:



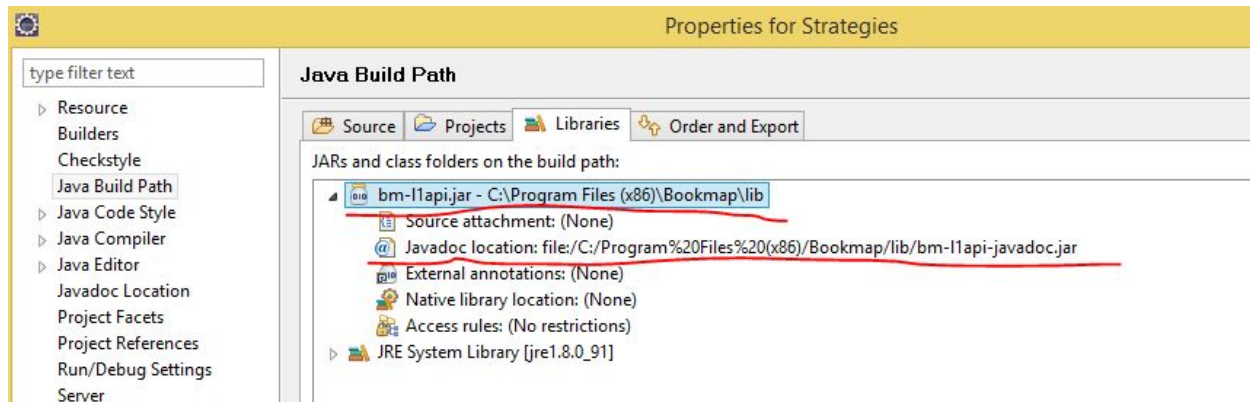
If you don't see it - please send us your license key and we'll check if strategies API is enabled for your license.

This is the strategies button and bookmap already has some built-in strategies. When you open it you can enable/disable/configure those. Note, that sources for few of those are available as a part of a project available at <https://github.com/BookmapAPI/DemoStrategies>.

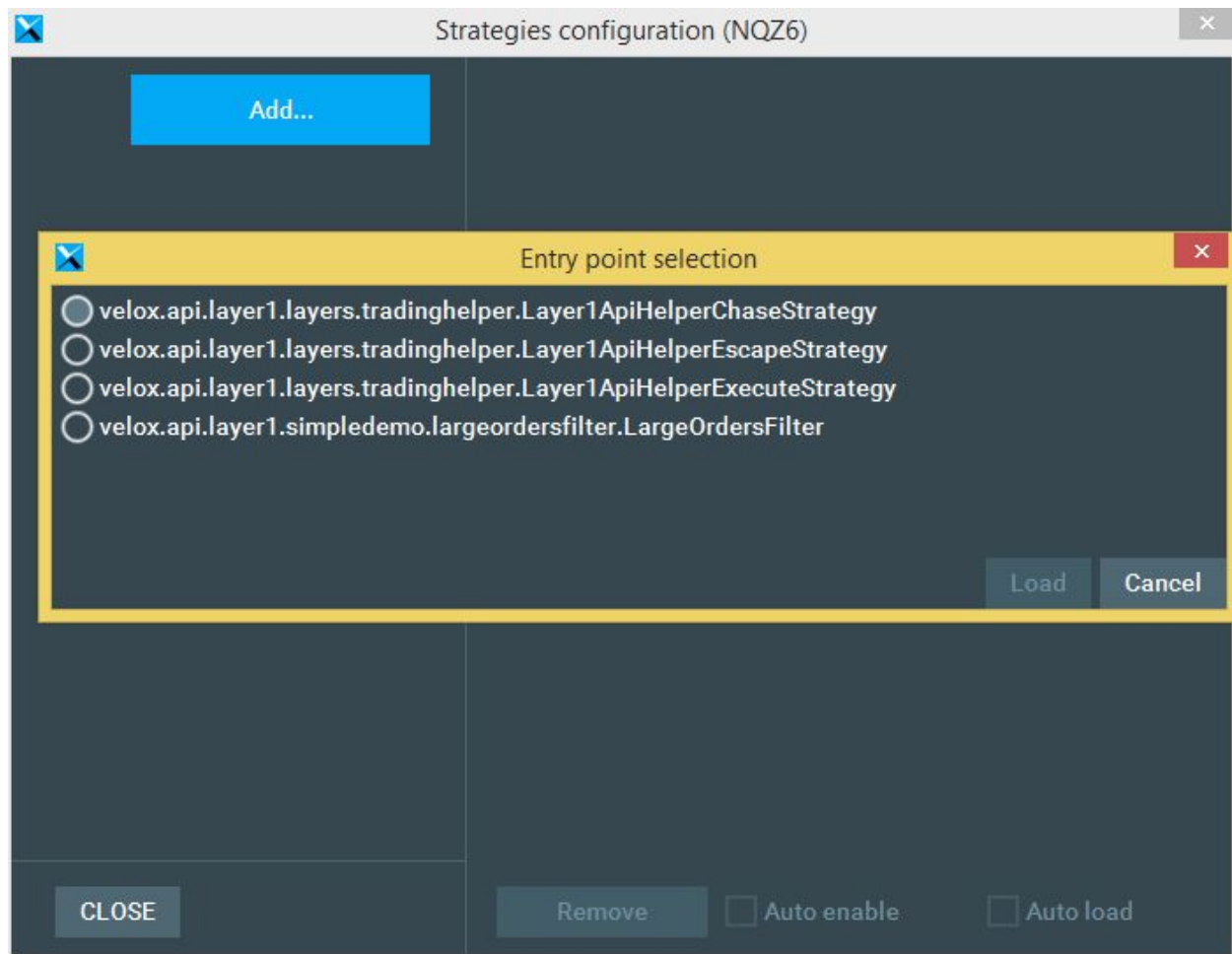
Now, since you will probably want to edit those strategies to gain understanding of the API, let's remove embedded version from bookmap. For this just close Bookmap, go to C:\Program Files (x86)\Bookmap\lib and delete "bm-strategies.jar" and start it back. Button should remain where it was, but in a window that opens when you press it most (all) strategies should disappear.

Now you will need Eclipse for Java to develop your strategy. Note, that you need to use 32 bit JRE to run Bookmap with your strategy from Eclipse. Technically any IDE will do, but I'll use Eclipse as an example.

Import the project into workspace. If you see errors please check that you have build path of the project configured correctly:



These settings should be there out of the box, but if not - please configure it. Now you should be able to export the strategy as a Jar file. Please do it and try to import that jar into Bookmap. You should see this window:



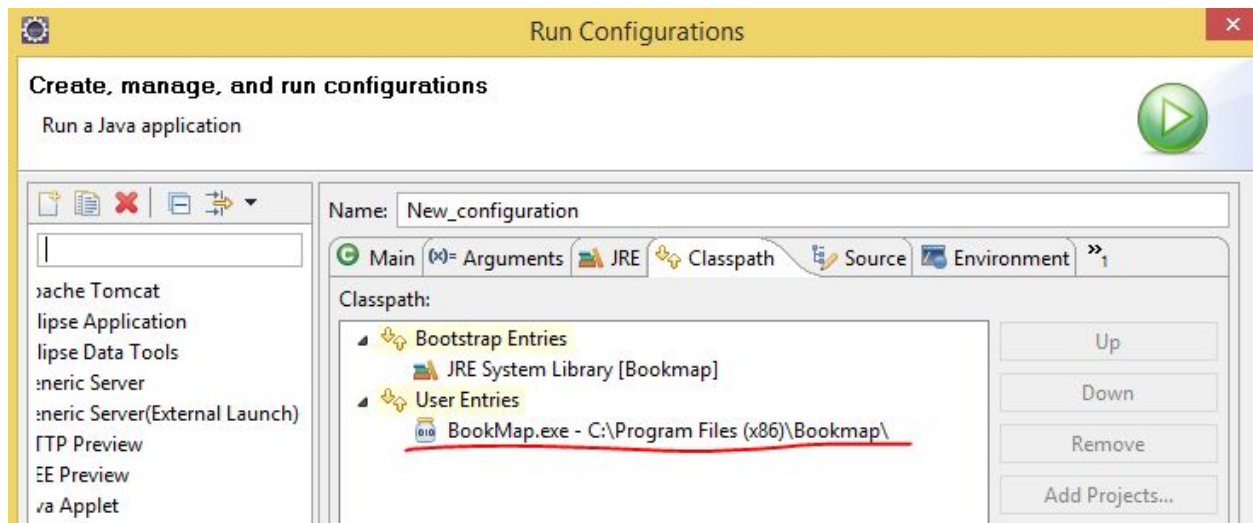
You should see a list of entry points, each one represents a strategy. Some are the ones you've already seen.

You can develop by editing the project and importing new jar files, but it's not very convenient, so here is how to make it easier

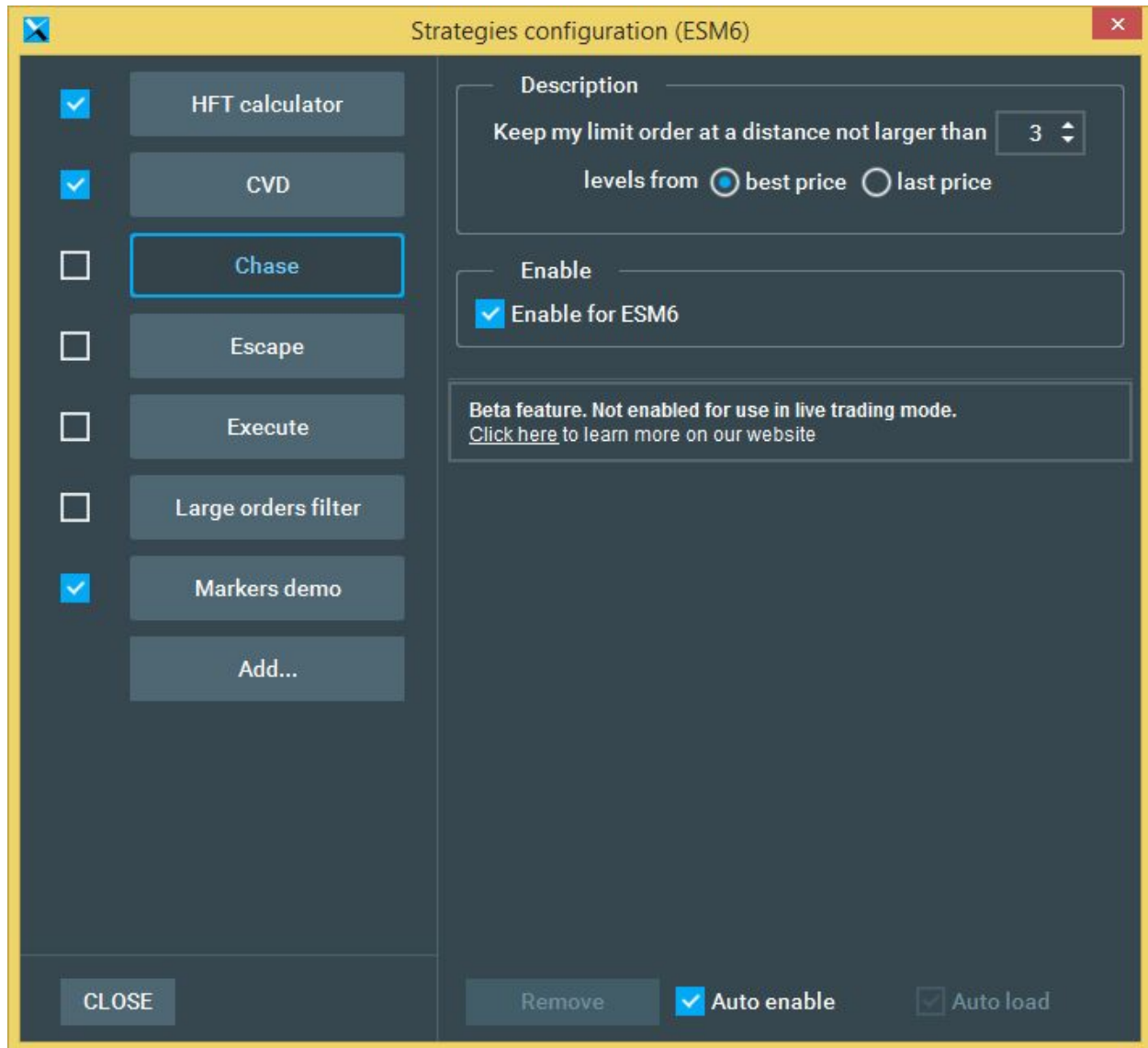
(this is described in velox.api.layer1 package description, but I'll copy it here with some remarks):

- 1) Create empty "bm-strategy-package-fs-root.jar" in the folder with your classes (For Eclipse it's <your workspace>\Strategies\bin).
- 2) Select it in bookmap strategy loading dialog and configure bookmap to load the strategy (e.g. pick the last entry point, make sure "Auto enable" and "auto load" is on and strategy is enabled in bookmap). Now you can remove the file.
- 3) Configure you IDE to run bookmap. For this configure ide to
 - add C:\Program Files (x86)\Bookmap\Bookmap.exe to classpath
 - start velox.ib.Main
 - Add -Xmx1000M to VM options
 - Use C:\Bookmap\Config as working directory
- 4) Now you should be able to use this configuration to start bookmap with latest version of your strategy.

Note that JRE used to start bookmap should be a 32 bit one (you can use a bundled one from C:\Program Files (x86)\Bookmap\jre) and Bookmap.exe should be added as User entry (in Eclipse run configuration):



When you start Bookmap this way you will be able to debug your strategy from IDE. Here is the screenshot of what you should be able to do (I've added Strategies project to source lookup when Eclipse asked for it) (to repeat it place breakpoint into the same place and with "Large orders filter" strategy loaded as explained above send an order of size 11 or larger - it will be "rejected" by strategy and if you are in debug mode you will be able to control it step-by-step):



See examples for details.

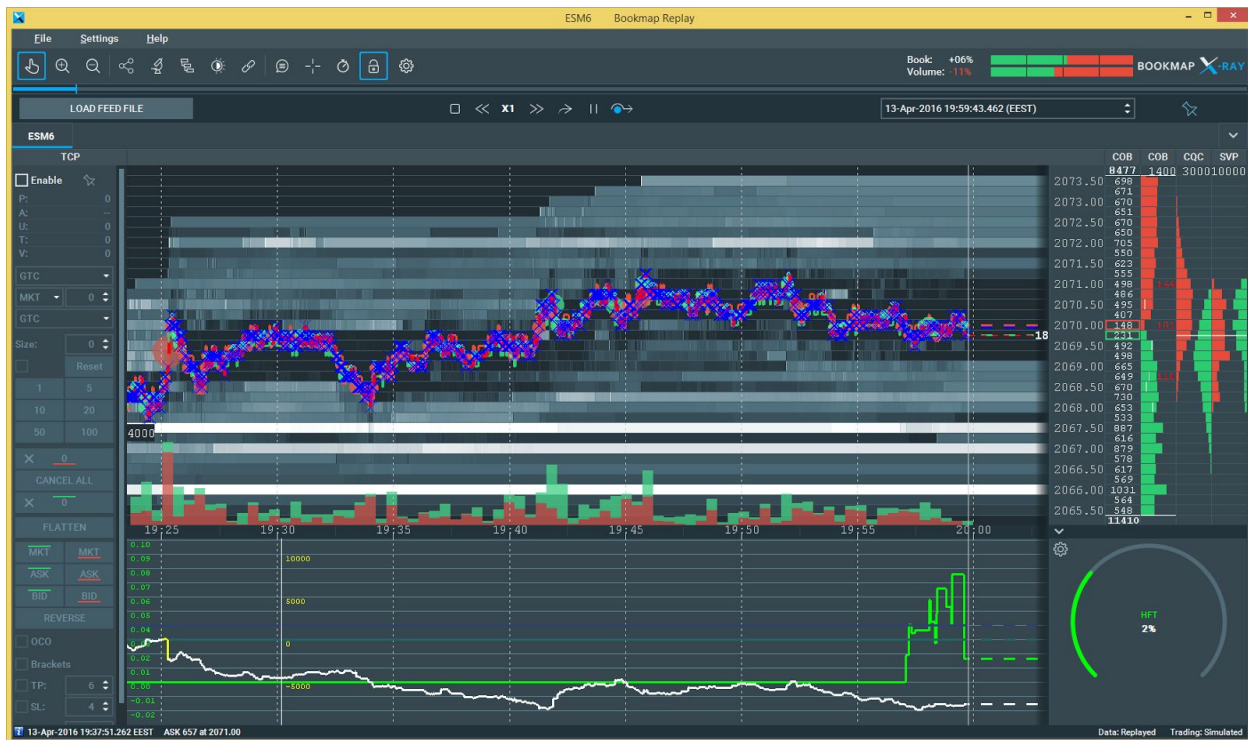
Feel free to contact us if you need some functionality that is not implemented.

Indicators

As a part of bookmap L1 api you can develop indicators that are displayed over the heatmap or in the panel below it. Examples are still a best place to get started, but here are few general ideas:

- Indicators are just a part of a strategy. Any strategy can register any number of indicators.

- It can be displayed over the main chart or over the bottom panel. Also indicator can optionally be displayed as a widget:



The ugly red line with many blue X marks is a demonstration of drawing over the heatmap and the lines in the bottom are the bottom panel indicators. The widget is in the right bottom corner.

The strategy that uses widgets is required to implement `OnlineCalculatable` interface which contains two important methods:

- `calculateValuesInRange` - this one will be called when your indicator needs data for a specific time range. Usually it happens when user moves heatmap around. This method does not necessarily have to be very fast - if you need to make complex computations you will be able to provide values one-by-one and user will see the computed part of the graph.
In this method you will usually request aggregated data from a storage and process it.
- `createOnlineValueCalculator` - this one is intended for calculating indicator values in real time. You will have to provide a calculator that will react on incremental updates recomputing the last value as necessary.