User's Guide of the CROSSMINER Advanced Integrated Development Environments Plug-in

FrontEndART Szoftver Kft.

Version 1.0.80.rev3 • March 18, 2019

Chapter 1

General features

In this chapter we present several functions which can not be linked to any single feature. These embodied an overview about the general principles which interwove the whole plug-in.

1.1 Installation guide

Update sites are used to organize and export features so they can be installed into Eclipse applications. A feature is used to package a group of plug-ins together into a single installable and updatable unit. Features have a manifest that provides basic information about the feature and its content. content may include plug-ins, fragments and any other files that are important for the feature. A feature can also include other features. The delivery format for a feature is a JAR, but each included plug-in will be provided as a separate JAR.

1.1.1 Create an update site with Eclipse IDE

First of all, you need to create a feature, from our plug-in. You can create a new feature by creating a New Feature Project. After your feature project is completed, you have to add our plug-in in feature.xml Included Plug-ins tab as Figure 1.1 shows.

When your feature is ready, you have to create an update site. You can create a new update site by creating a New Update Site Project. After your update site project is completed, you have to add the feature in site.xml Site Map tab as Figure 1.2 shows. After you added the feature to site.xml you can Build an update site.

1.1.2 Install plug-in via an update site

To install plug-in via an update site select the *Install new software* under the Help menu in Eclipse. As you see on Figure 1.3 the following window

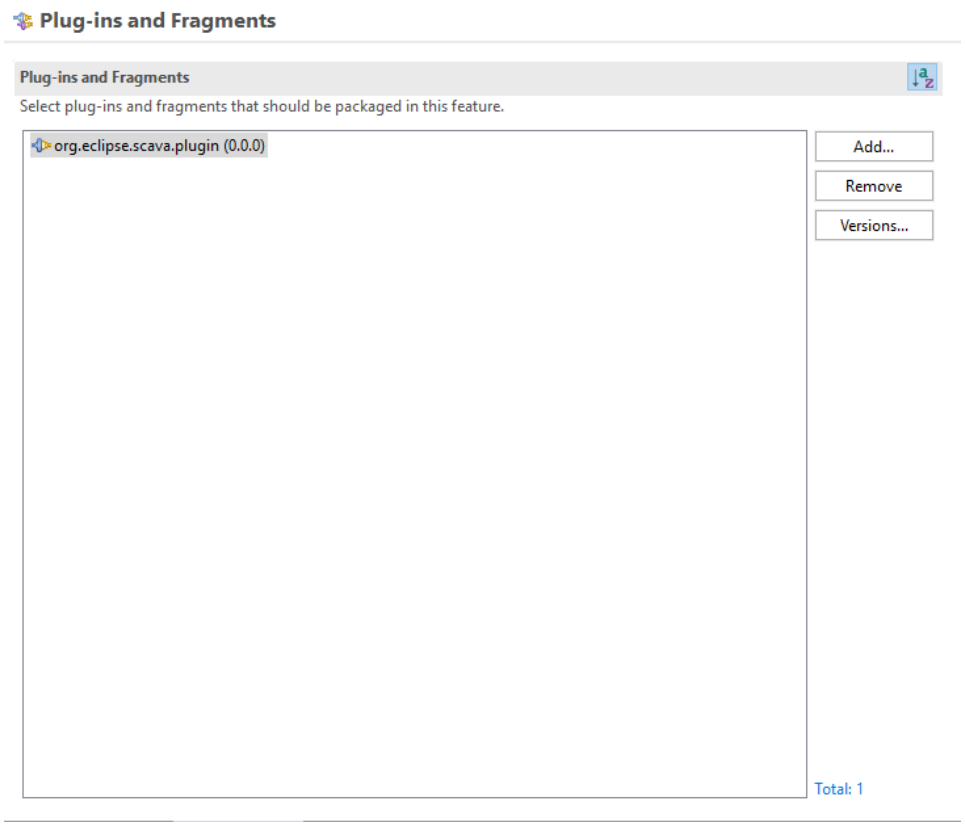

Figure 1.1: Included plug-ins window

you can choose the location of the update site. The location can be a remote or local one. After you select the update site's location it will be shown in the list. You can browse the plug-in's features and disable it if it is not necessary for you. After you are ready, click the *Next* button. On the following screen you have to accept the license agreement if you want to continue the installation. The last screen you see the installation details, on this screen there is a list which contains all of the selected feature for your installation. To finish the installation process click on the *Finish* button.

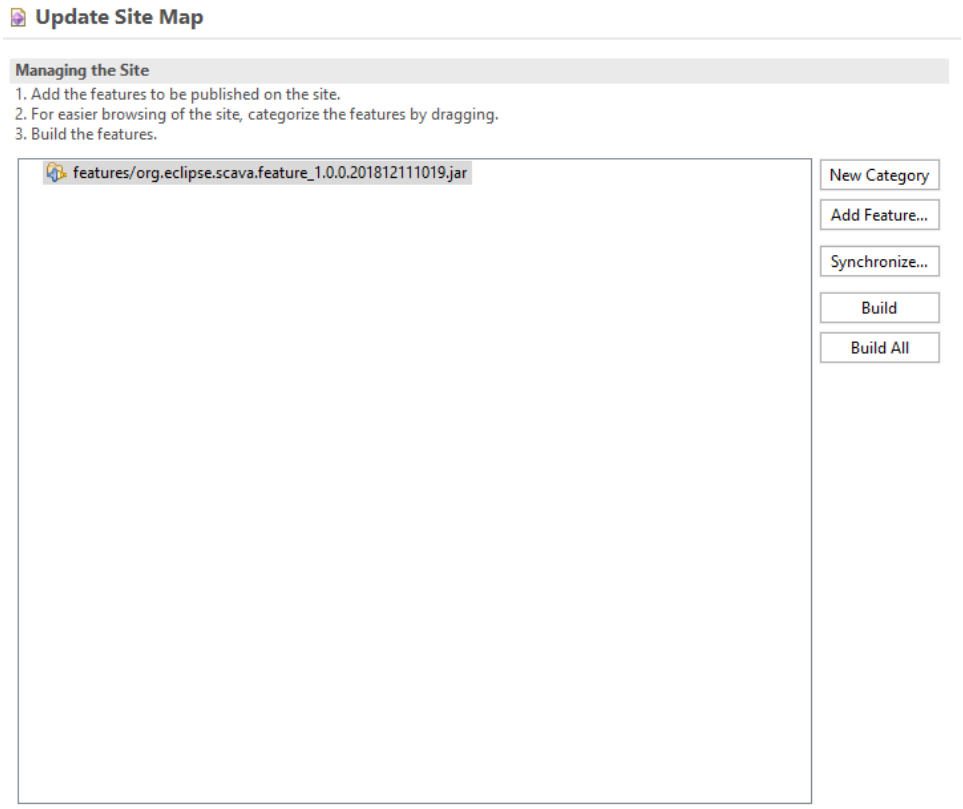

Figure 1.2: Site Map window

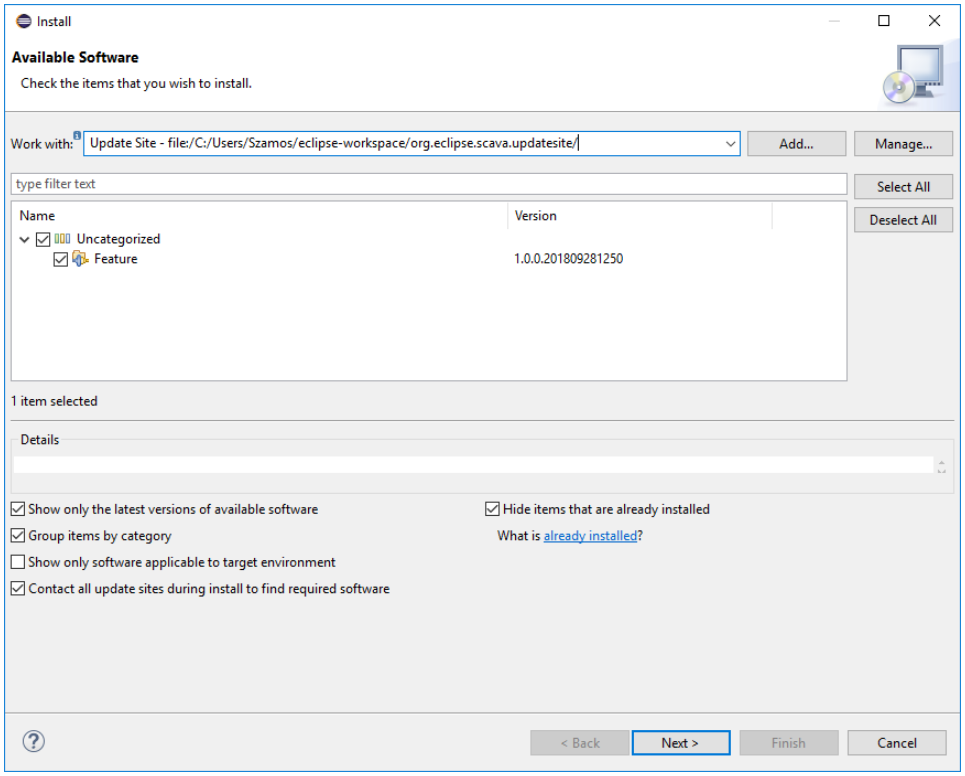

Figure 1.3: Available Software window

Chapter 2

Acquire Recommendations

There are several kind of suggestions and recommendation which could be helpful for the developer during their daily tasks. In this chapter we introduce those features which result some kind of recommendation. One of the main properties is the type of entity which are the subject of the suggestion. To simplify any further discussion our tool only permit recommendation with a single subject type. You can find more details in the following sections.

2.1 Library Based Recommendations

Recommendation which subject are libraries are called *library based recommendations.* This feature helps you to search libraries whiches fit in your development process.

2.1.1 Searching Additional Libraries

One sub-type of library based recommendation when the goal of the developer is to find new libraries based on some user specified search criteria.

To use this feature select the Library Search item under the CROSSMINER tab in the toolbar. To search differenct libraries, which are needed for your project enter its name (or part of it) and click on the *Search* button. After a succesful search a list of libraries will be shown under a new tab. A list of libraries which meet the requirement will be shown on the left side of the window. You can easily switch between them. You can run different searches which are open a new tab. To switch between them click on the tab header. You can close this result with the cross on the tab header if you no longer want to use it. You can see on Figure 2.1 the user search for json library. The results are on the left side of the window and central panel holds information about the selected library.

To get information about the libraries select the list item. After selection the central panel shows information about the libraries and provides

weblinks to the library GitHub repository and the Crossminer WebDashboard. The link opens webpage in your default browser, if you did not set up default browser an error message will be shows up. If you find the right library you can simply add it to the installer queue just by clicking on the *Add* button under the library name. The libraries which are selected for the installation will appear on the right side of the window. To start the installation click on the *OK* button.

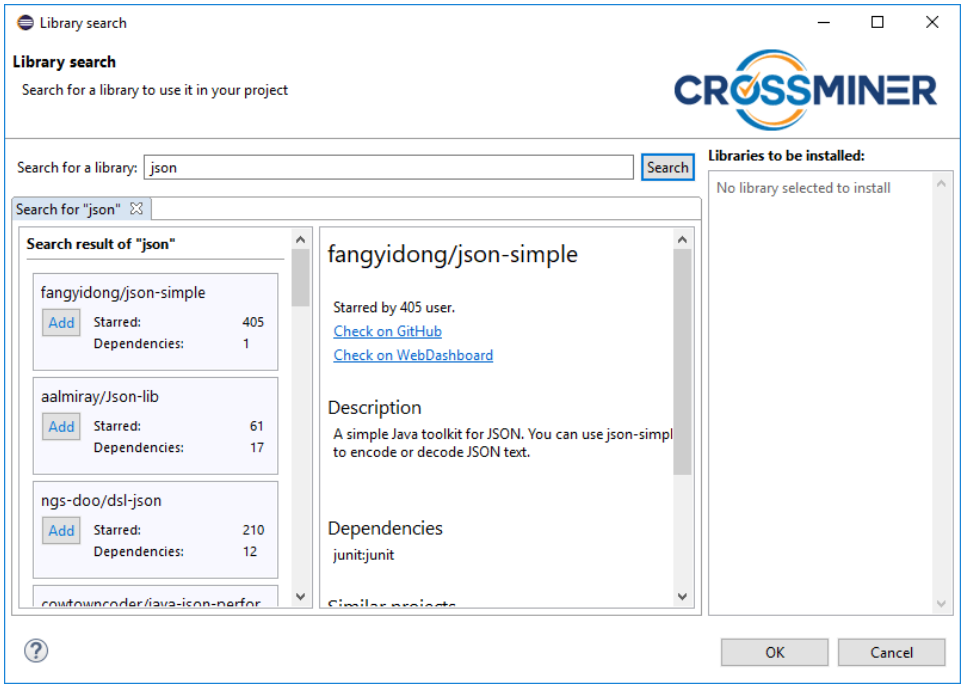

Figure 2.1: Library search window

2.1.2 Searching Similar Libraries

There are usually some pre-existing selection of libraries which serve as a base to find more relevant API-s. We split these kind of recommendations into two major sub-categories: whether the pre-existing set of libraries are already installed or not.

Suggestion Based on Non-installed Libraries

If you want to search libraries not by its name but its funcionality use search similar libraries function. This function placed on the bottom of each library details. Select a library and scroll the description all the way down. On Figure 2.2 user select simple-json library by fangyidong which is a well-know json parser library. The user also searched library which are

similars to this. To search similar libraries like the selected one click *Search similars with method* link on the bottom of the information page. This method will open a new tab where you can browse among similar libraries.

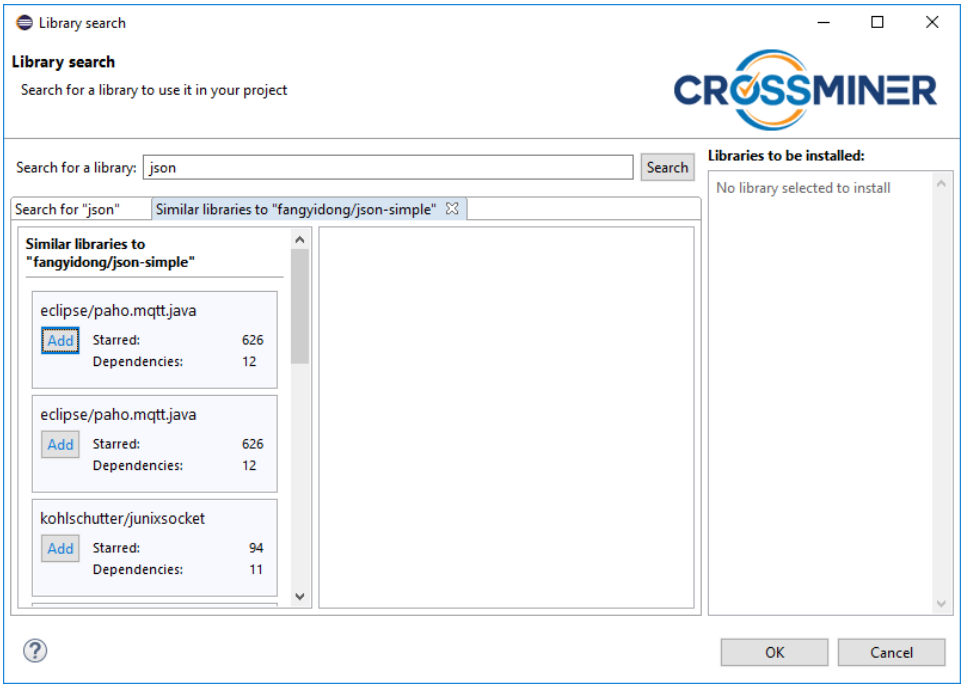

Figure 2.2: Similar libraries like the selected one

Suggestion Based on Already Installed Libraries

Before the plug-in requests recommendation about how to installs the selected libraries you can search for additional libraries based on third party APIs already present in you Maven settings. The interface offers a special list of libraries which is based on the previously installed ones. If your project isn't a Maven project, this feature is not work for you, the plugin process the Maven Pom file to detect installed libraries. You can select which libraries will be used during the search. To select relevant libraries use the checkboxes. On Figure 2.2 user have three installed libraries and select that all of them is relevant to similiarity search. To search additional libraries click on the *Search for additional libraries* button. This action opens a new tab which is similar to the previous search results view, but this library list contains those libraries which are similiar to the currently installed ones. You can add libraries to the list the same way as you did it before. To finalize installation process select the *OK* button and the plug-in will install the selected libraries.

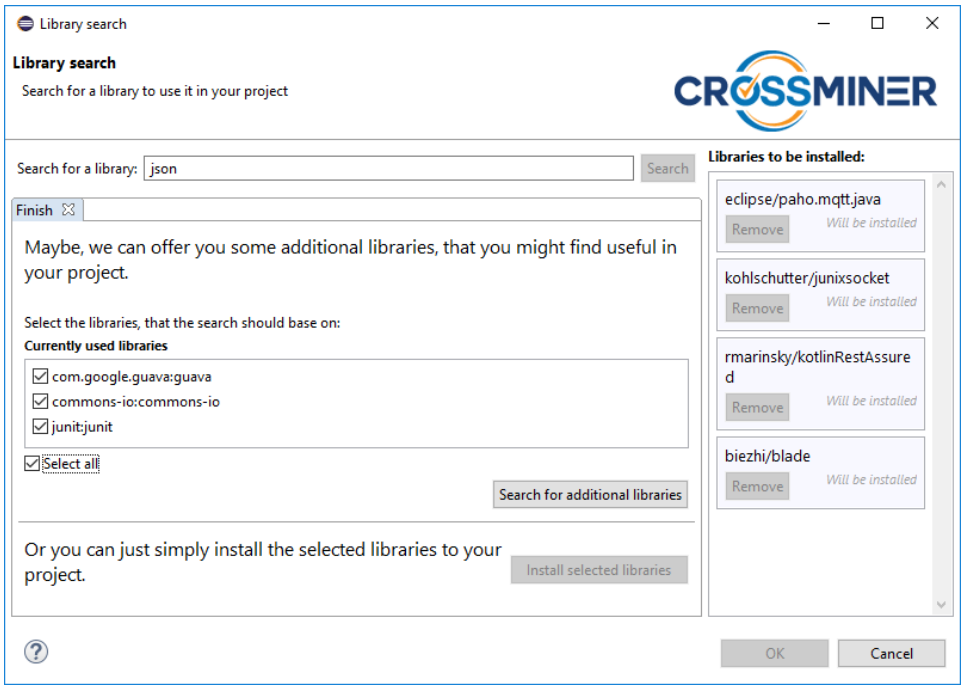

Figure 2.3: Library search finish

2.1.3 Handling Changed and Deprecated APIs

Third party libraries are prone to change and evolution. To help the developers adapt their project to these changes we defined an other sub-category of recommendations, namely when their goal is to provide information about modified or deprecated interfaces.

2.2 Source Code Based Recommendations

In this section we elaborate features related to those recommendation which subject entities are present in the source code of the project under development. They usually retrieve some code chunk, which could be annotated to ease further understanding.

2.2.1 Retrieving Suggested Code Snippets

Based on the current development context, our plug-in is able to yield a set of source code snippets (code chunks), which could be useful to implement or to understand various features. This feature helps to improve your code quality based on your suggested code snippet. The plug-in able to give you a recommendation how to change your code to improve it. To use this feature

select a code chunk and press the right mouse button. In the context menu select the *Request Recommendation* under the CROSSMINER entry.

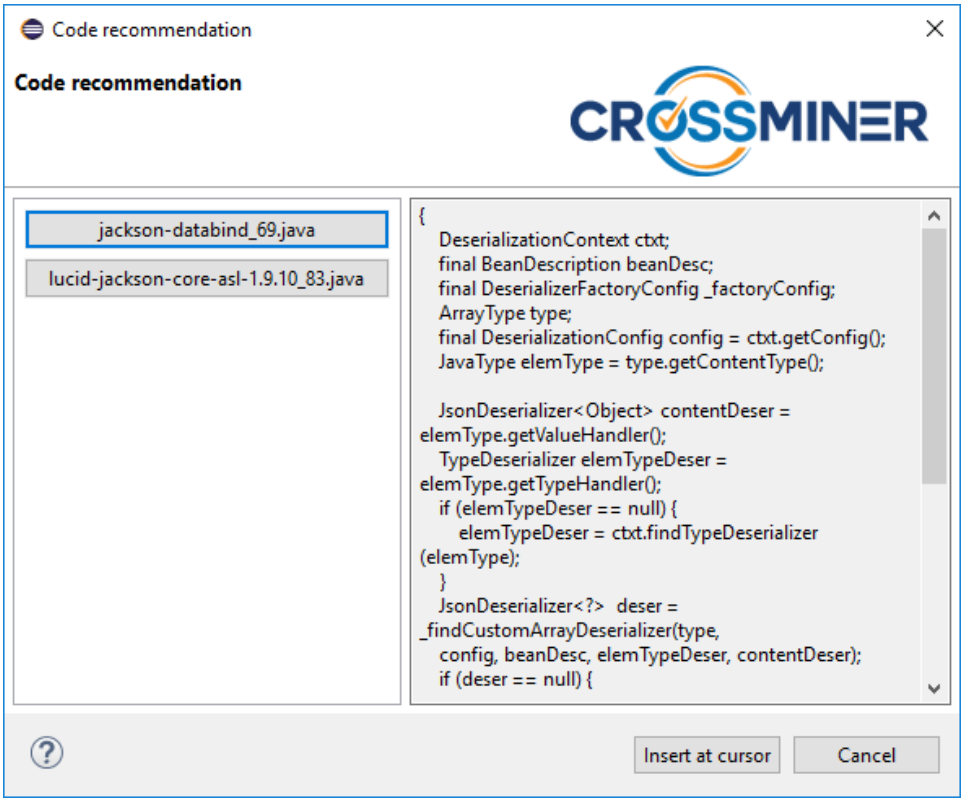

Figure 2.4: Code recommendation view

As shown on Figure 2.4 the plugin (in this example) suggest two code improvements for your selected code chunk. You can browse them and inspect the source code. To insert the selected source code click on the *Insert at cursor* button. This action will insert the code at your current cursor location and if there is a selected code chunk then it will be replaced. The recommended source code is a pattern, it only helps you to understand the logic but it is not guaranteed to be semantically (or in rare occasions, syntactical) correct.

2.3 Text Based Recommendations

Finally there are tons of documentation and discussion available for various topics, which could be useful for the developers. Those recommendations which yield some natural language documents (or reference to them) are called *text based recommendations.*

2.3.1 API Usage

Developers could use these recommendation to get more information about the features and their usage of a 3rd part API, for example official pages, documentations, and samples.

2.3.2 Handling API changes

There several forum threads and change reports, which could ease the migration between different versions of the same library.

2.3.3 Inspecting Code Chunk Related Q&A Posts

Also there are some discussions about how to implement a function using a specific features or a set of libraries. The recommendations which present these documents (posts and blog entries) are represent an other subclass of text based recommendations. To get relevant posts about your code, select it and press the right mouse button and click on the *Request API documentation or QA post* in the CROSSMINER sub-menu. One similar panel as Figure 2.5 will be showed at the bottom of Eclipse. This panel contains a list of available QA post for your selected code chunk. For further information you can open the listed links (by clicking on them) in your default browser.

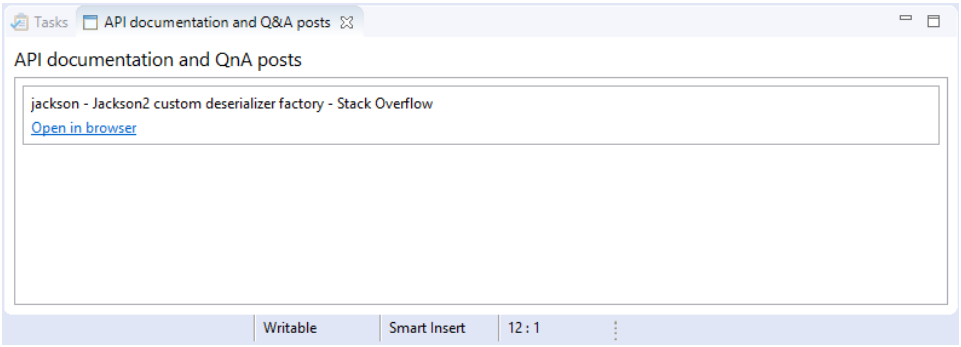

Figure 2.5: Code recommendation view

Chapter 3

User Activity Monitoring

The main components of this scenario is the recording of developer interactions (events), computing process related metrics and sending them to the CROSSMINER server for further processing. The collected events is stored in a local database. Please note that none of the stored event contains information about the user, so there is no way to identify someone from these data, also all of the events are stored in a local database and only the metrics are sent to the CROSSMINER Server.

3.1 List of Collected Events

Document event Our plug-in using it to detect all keypresses in the editor and store which file is affected. For example during the implementation of a new method for an existing class.

Part event It stores information about life cycle of given parts (part activating, part deactivating, part closing etc). For example when you work on a existing class and select something in the Package Explorer.

Window event It stores information about window life cycle the same way as the Part event handles information about parts. For example when you open an another program.

Eclipse close event Eclipse closes are stored in this even type.

Launch event It stores information about code building and launching.

Resource element event It stores information about the file saving and deletion.

Class path event It stores information about class path changing (add entry to class path or removed entry from class path). For example when you add a library to your clashpath.

CROSSMINER event These events are invoked by using plug-in. It needs to be calculated for those metrics which measure the plug-in usage. For example when you use CROSSMINER Library Search function.

3.2 List of Recorded Metrics

3.3 Retrieving Process Metrics via crossminer Web-based Dashboard

You are able to inspect any computed process metrics for your project by using the relevant features of CROSSMINER Web-based Dashboard.

3.4 Plug-in Side Debugging Features

In the case of unexpected errors during the user activity monitoring, you are able to check the value of the process metrics and some relevant meta-data about the underlying database on the client side. To do this please activate some of the plug-in side debugging features. Please note that these are only available in the debug version of the plug-in.

3.5 Server Side Debugging Features

There are ways to access the raw data, stored at the server side. To do this you have to execute the following REST API calls. You could use your preferred REST client, but for illustration purposes we will use Postman[1]

[1] `https://www.getpostman.com/`

Chapter 4

Settings and Customization

All the plug-in related settings can be found in the plug-in's preference page. To open this page select the Preferences item under the Window menu in Eclipse menu.

4.1 Integration Related Settings

To properly use the plug-in, you have to set some of the settings which is required for integration. As Figure 4.1 shows you have to enter the Knowledge Base server address, a port and the Web Dashboard's base path.

4.2 Process Metric Related Settings

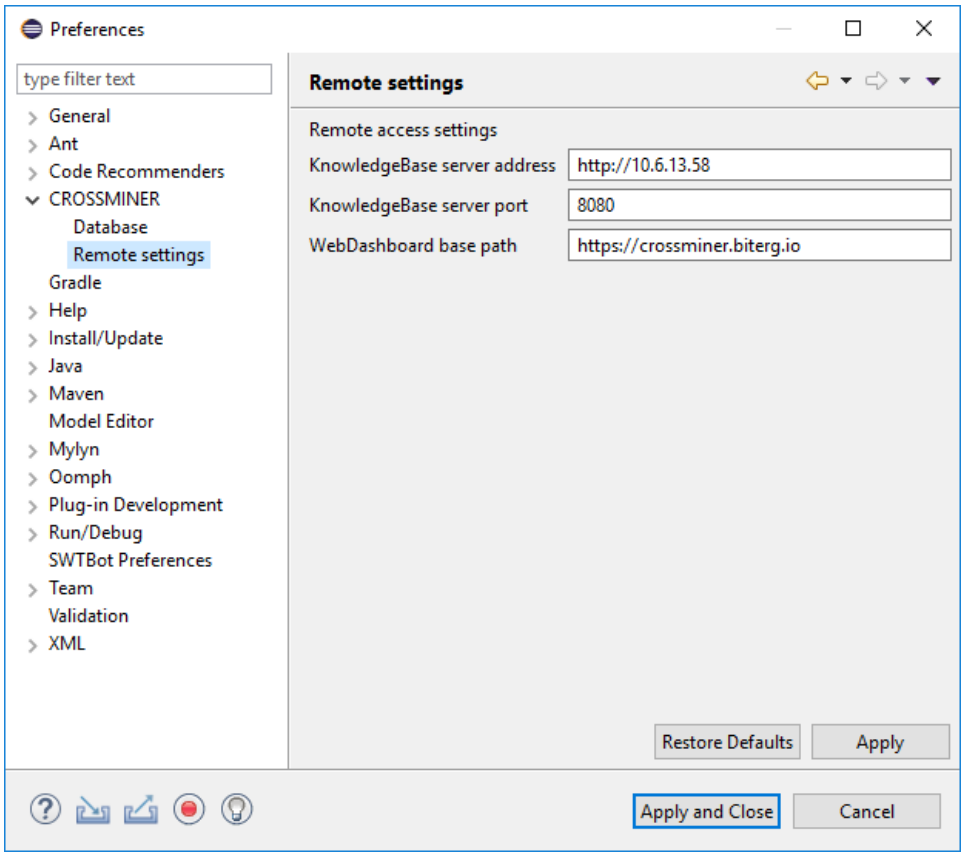

Figure 4.1: Plug-in remote settings