



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

FrontEndART Szoftver Kft.
Version 1.15.0.rev0 • October 24, 2019

Contents

1	General features	5
1.1	Installation guide	5
1.1.1	Create an update site with Eclipse IDE	5
1.1.2	Install plug-in via an update site	5
2	Acquire Recommendations	9
2.1	Library Based Recommendations	9
2.1.1	Project search	9
2.1.2	Searching additional libraries	11
2.1.3	Updating libraries	12
2.1.4	Handling Changed and Deprecated APIs	16
2.2	Source Code Based Recommendations	16
2.2.1	Code recommendation	16
2.3	Text Based Recommendations	18
2.3.1	API documentation and Q&A posts	19
2.3.2	API Usage	20
2.3.3	Handling API changes	20
3	User Activity Monitoring	21
3.1	Database settings	21
3.2	User Activity Monitoring Settings	21
3.3	List of Collected Events	21
3.4	List of Recorded Metrics	23
3.5	Retrieving Process Metrics via CROSSMINER Web-based Dash- board	23
3.6	Plug-in Side Debugging Features	23
3.7	Server Side Debugging Features	24
4	Settings and Customization	25
4.1	Integration Related Settings	25
4.2	Process Metric Related Settings	25

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

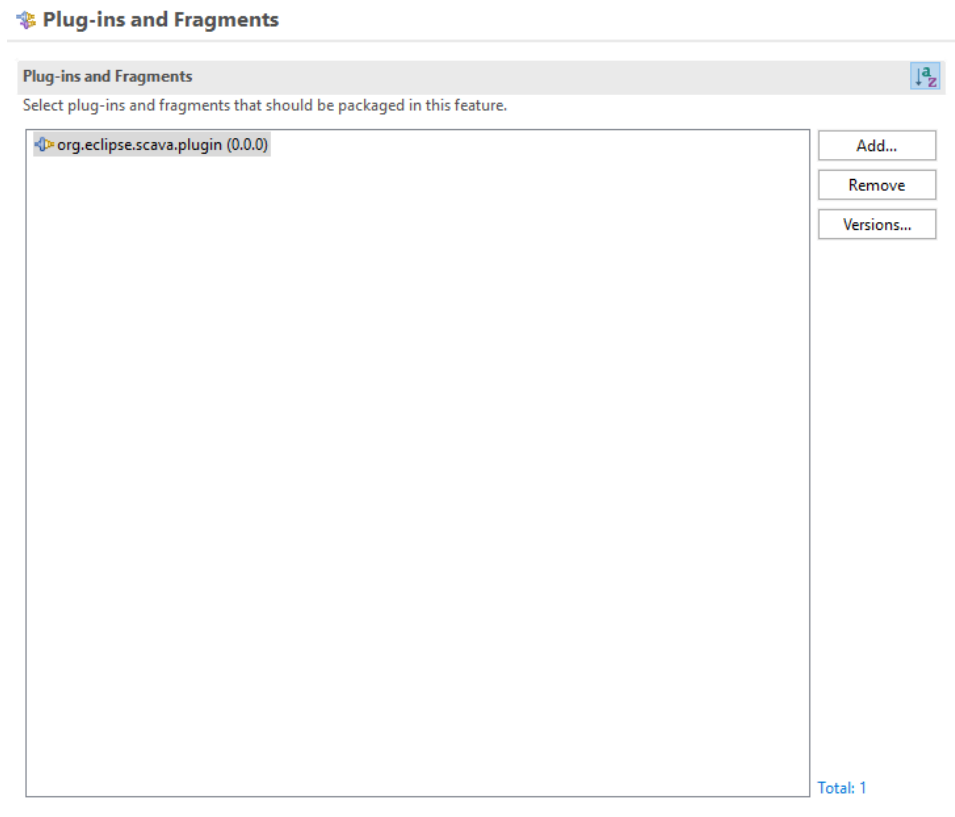


Figure 1.1: Included plug-ins window

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.

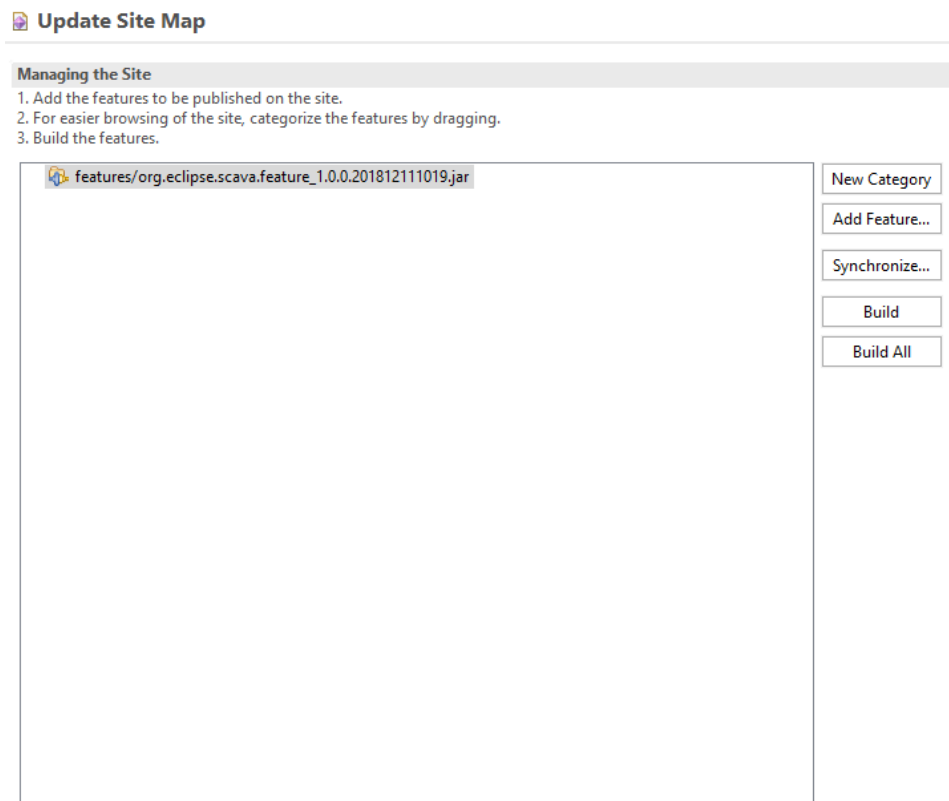


Figure 1.2: Site Map window

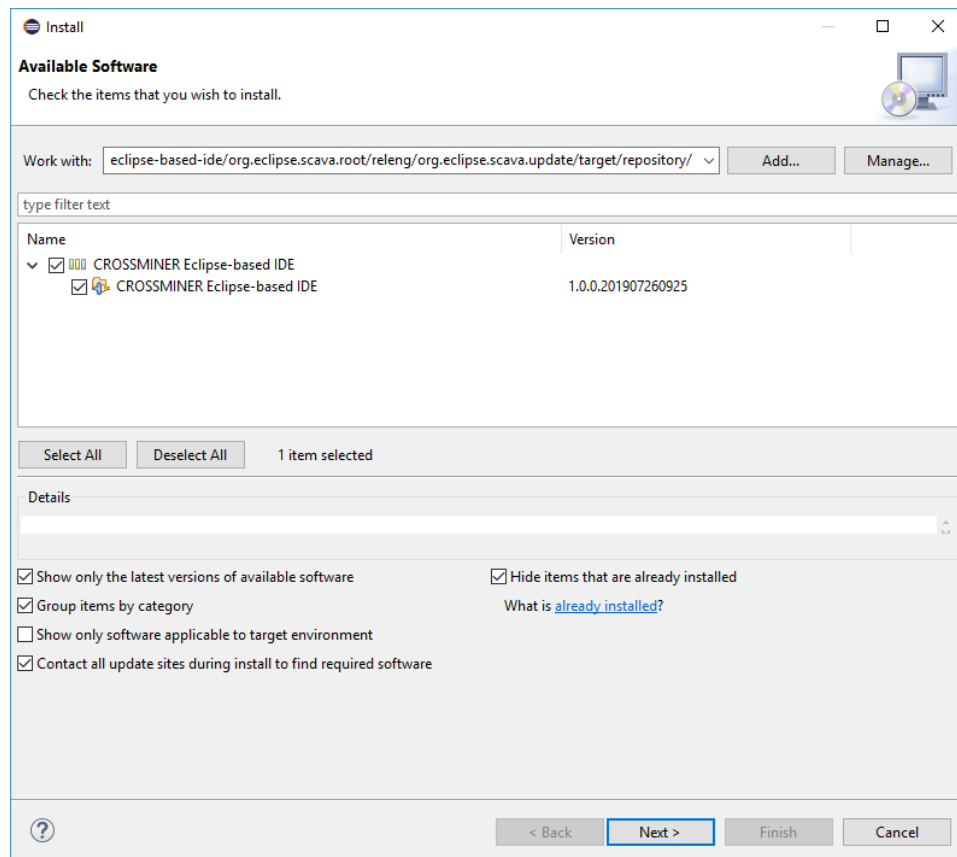


Figure 1.3: Available Software window

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 which fit in your development process.

2.1.1 Project search

To search for projects which can be installed into your system, select the Project Search feature under the CROSSMINER menu. After the dialog opened, you can type the desired search expression into the search field. After that, hit enter or click on the Search button. The results are going to show up in a new tab in the middle of the dialog. Every search is going to open a new tab and show up inside that.

On the left part of this tab you can see the results list. Initially only a few of these results are shown in the list, but by clicking on the Show more button at the bottom some more results can be requested. If no more results can be loaded then the button will no longer be available and a short text will inform you about it. By clicking on these items a couple of informations are shown in the right part of the search tab, like a description or the dependencies. You can navigate to the GitHub repository, to the official website of the project or to the related CROSSMINER Web-based Dashboard site. You can initiate a new search for similar projects to the

given project by clicking on the similarity method buttons. These are the Compound, Crosssim, Dependency, Crossrec, Readme, RepoPalCompound and RepoPalCompoundV2. Or you can select the given project result to be installed by clicking on the Add button in the result's box. The selected projects are shown on the right side of the dialog in a list. By clicking on these items you can view the details of them such like in the previous case. You can deselect these items by clicking on the Remove button. After you chose the wanted projects to be installed click on the Install button on the bottom of the dialog. The search dialog can be seen on the Figure 2.1.

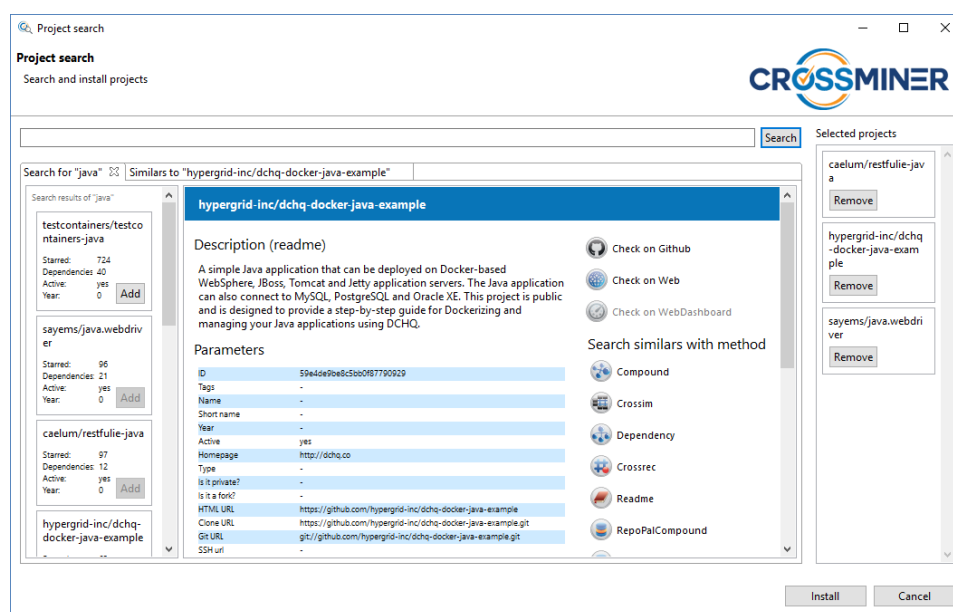


Figure 2.1: Project search dialog - search

In the install dialog you can set the base path to the location where you want to install the projects by default. After updating this path all of the projects destination location is going to be updated. You can see this property of the projects on the bottom of the dialog, in the project list. All of the projects can be installed into different locations from the default by giving them a custom destination path. After setting all of the desired install paths, you can choose to install all the projects at once by clicking on the Install all button or you can install them one-by-one manually, by clicking on the Install buttons of the projects. The installation process can be canceled by clicking on the Cancel buttons. The install dialog can be seen on the Figure 2.2.

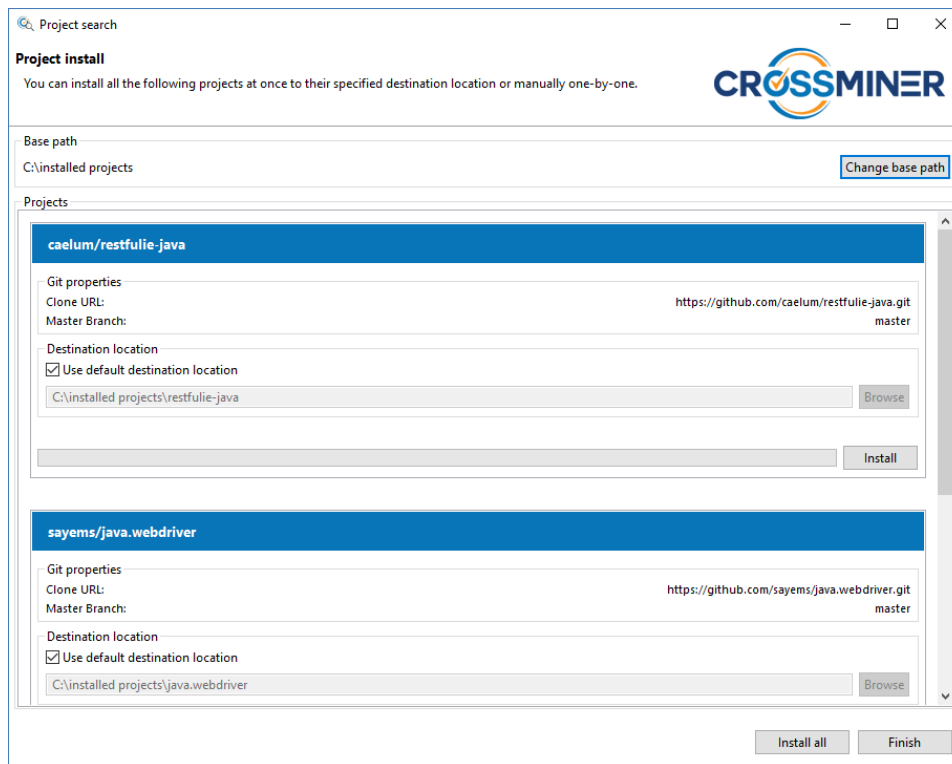


Figure 2.2: Project search dialog - install

2.1.2 Searching additional libraries

To search for libraries that can be useful during the development of your project, open the Library search dialog. For this, first you have to select the project in the Package explorer view. This feature works only with Maven projects, that contains their pom.xml file in their root folder. Then navigate in the menu to the CROSSMINER menu and click on the Search libraries option. This will bring up the Library search dialog.

This dialog is split into two parts. On the left side you can see the currently used libraries in your project and on the right side there are the suggested libraries. Both the left and the right side contains their libraries in two separate groups. On the left side, in the upper group there are the libraries that has been selected to be used as the base of the search. The suggested libraries may vary by the different set of selected libraries. On the bottom there are the libraries that has not been selected yet. On the right side, in the lower group there are the libraries that the CROSSMINER suggested for you to use in your project. The dialog at this state can be seen in the Figure 2.3. These libraries can be selected to be installed, and if they are, then they move to the upper group which contains the libraries that has been selected to be installed. This can be seen on the Figure 2.4.

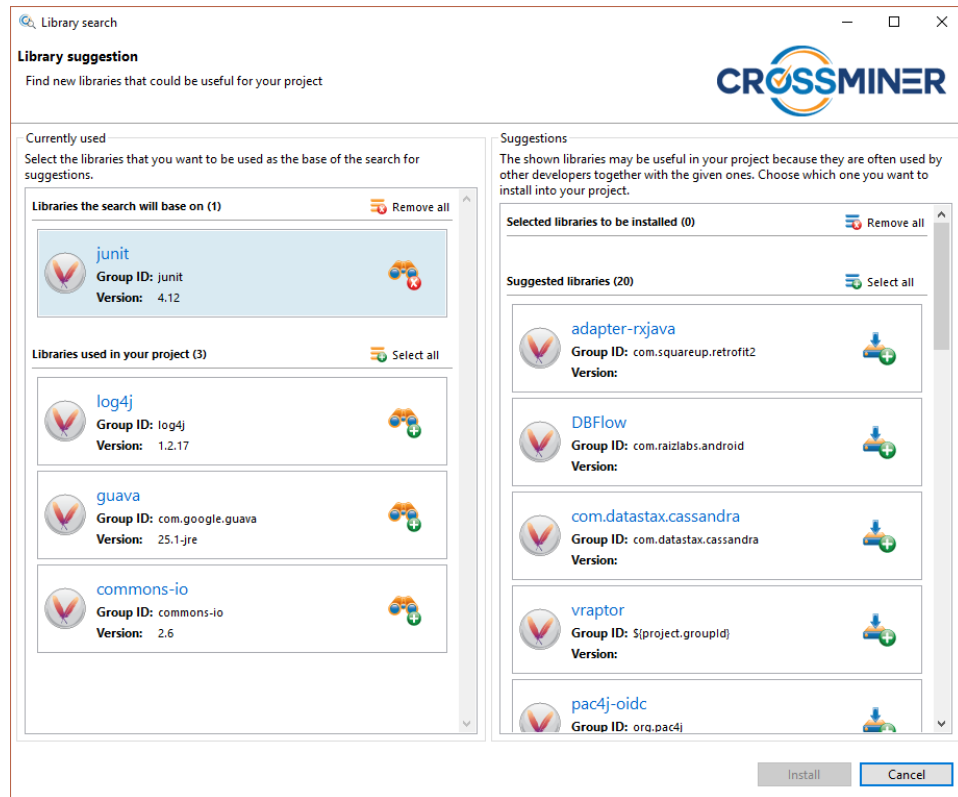


Figure 2.3: Library search dialog - results

Every time you change the search parameters, so the set of libraries that are used as the base of the search, the suggested libraries will be reloaded. If something goes wrong so the libraries cannot be retrieved from the CROSSMINER server, a Try again button will be shown and by clicking that you can initiate a new request to load them. This is shown in the Figure 2.5.

After selecting the set of libraries that you want to use in your project, by clicking on the install button you can add them to your project. They will show up among the dependencies in the project's pom.xml.

2.1.3 Updating libraries

If any of the libraries that are used in your Maven project can be updated to a newer version the CROSSMINER IDE will notify you by putting the "[new lib version available]" label after the project's name. An example can be seen on the Figure 2.6. This check is preformed at the start of the Eclipse IDE.

To update the libraries that are used in your Maven project use the Library version updater feature. For this, first you have to select your project in the Package explorer view, then in the context menu navigate to

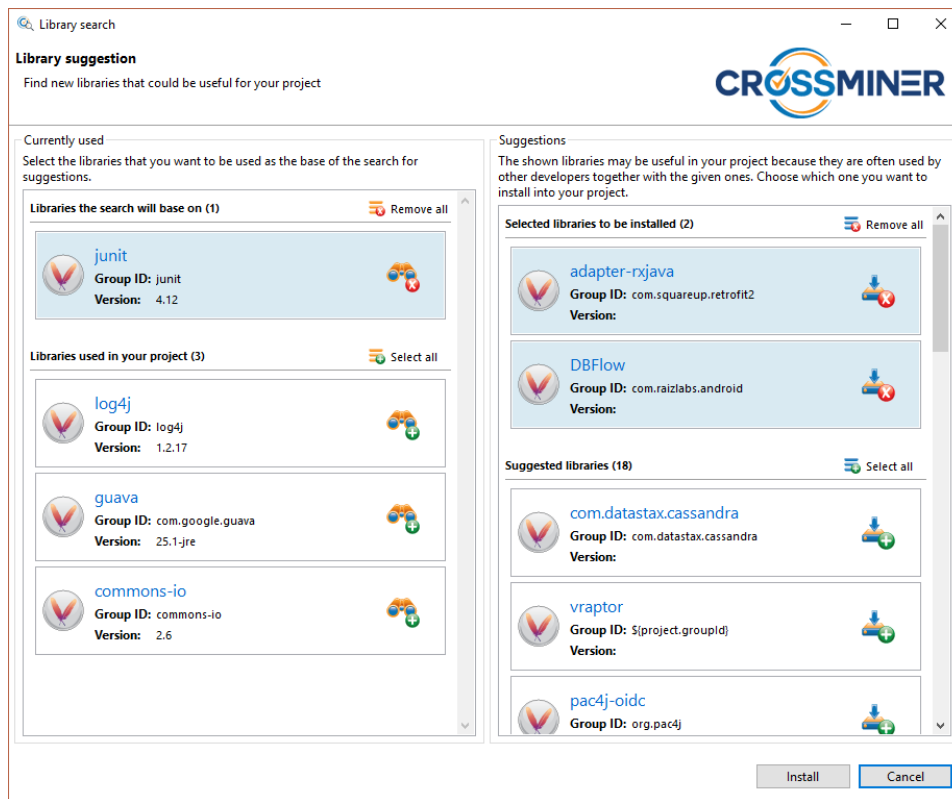


Figure 2.4: Library search dialog - selected to install

the CROSSMINER submenu and click on the "search updates for libraries" option. This will open the Library version updater dialog.

As it can be seen on the Figure 2.7 the libraries that can be updated to a newer version are listed in the center of the dialog. The different versions can be selected from the dropdown menus and the libraries will be updated to the selected versions.

The listed versions can be filtered by creating rules in the Library update version rules property page that can be accessed by opening the context menu over your project, clicking on the Properties option and navigating under the CROSSMINER pages.

Any number of rules can be added, removed or edited in this page. To create a new rule click on the Add new rule button. A box will appear in the center of the page with three text fields. These fields are the Group ID, the Artifact ID and the Version. Each field can be filled with regular expressions and they are interpreted just the same way as in Java. If a text field has been leaved empty it will be interpreted as to match with every value.

The rules can be removed by clicking on the "Delete rule" button in the

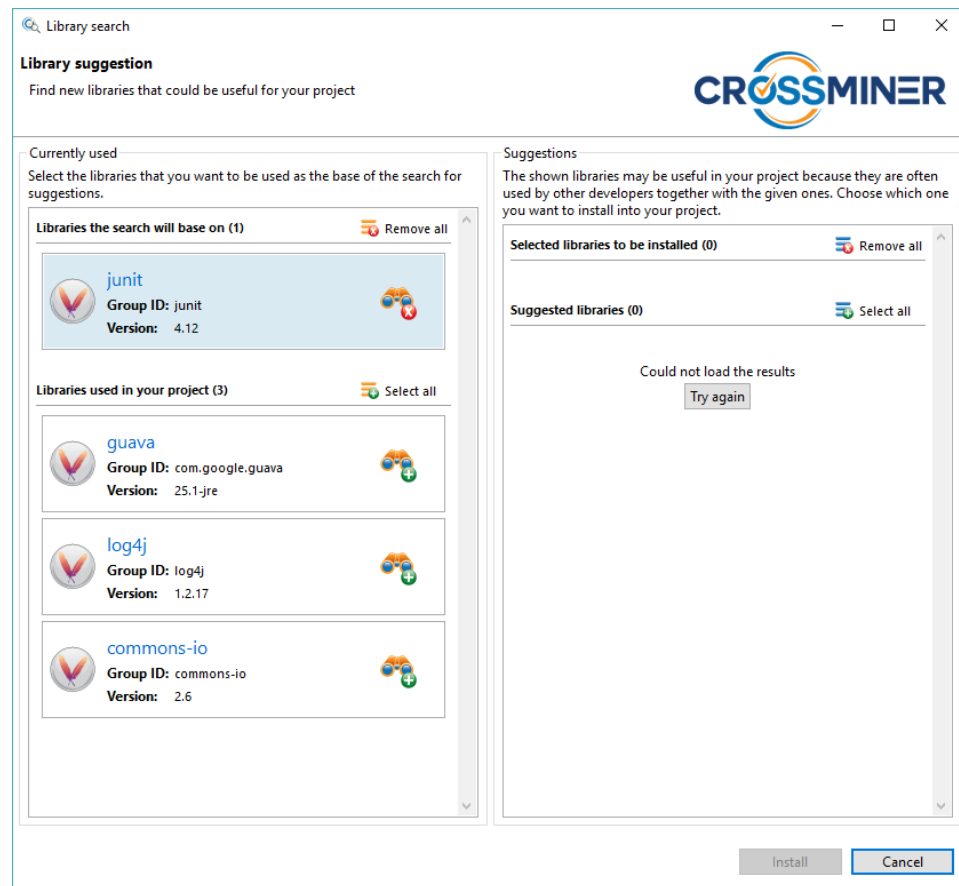


Figure 2.5: Library search dialog - try again

rule's box.

This property page and an example rule can be seen on the Figure 2.8. With this one rule we restrict the library version search to care only about the libraries with a group ID that matches with 'junit', an artifact ID, that matches with 'junit' and a version with a '4.' prefix.

The interpretation of the rules is the following: if there are no rules at all, then no filtering will be applied. After adding at least one rule, then the library versions will be tested against every rule until at least one of them is matching. If none of the rules matched a library version, then that one will not be seen among the available library versions. A filtered list of library versions with the previous rules can be seen on the Figure 2.9.

In the Library version updater dialog there are three buttons on the bottom, which can be used to quickly select the wanted versions. These buttons provide the functionality to select the currently used, the next available or the latest versions with a click.

After selecting the versions to be installed and accepting the confirmation

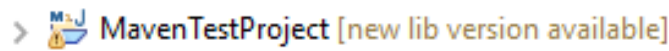


Figure 2.6: Library version checker

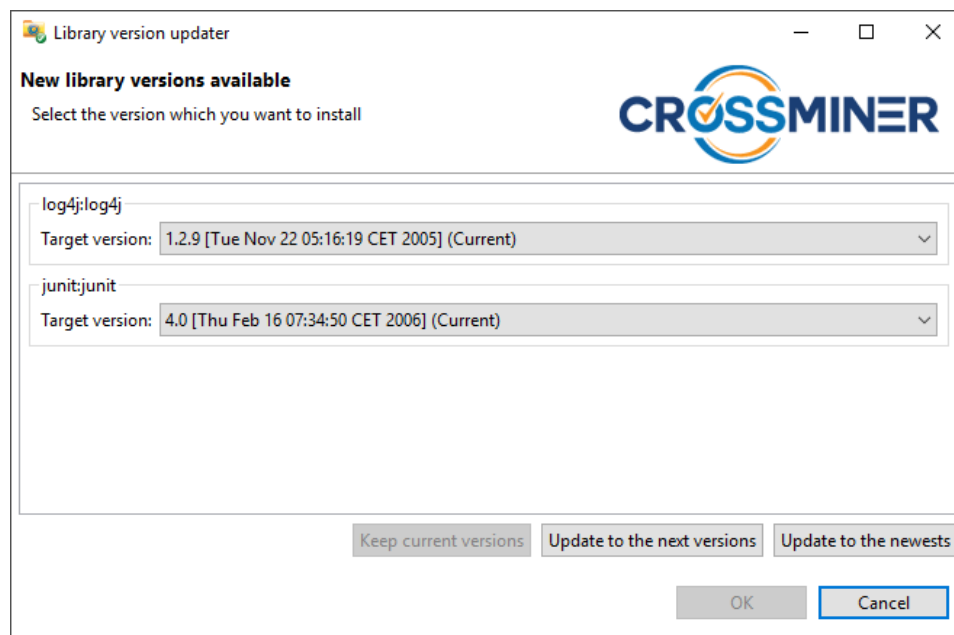


Figure 2.7: Library version Updater

dialogs, the selected versions of the libraries are going to be installed to your project.

If the upgrade from an older version of a library to a newer one involves some API braking changes, then the CROSSMINER Eclipse-based IDE will try to find the places of usage of the old API in the project and mark them as they are suspicious to be broken after the installation. The markers can be seen on the Figure 2.10.

Hovering you cursor over these markers the quick fix menu can be opened and the following options can be selected: you can choose the Fix option which will try to fix the broken part of the API usage near the marked code chunk. The Fix all and Fix all for API migration from old library to new library options will try to do the same but for every similar case in the whole project and to every similar case in the project, but only for those, that are related to the given API migration, respectively. The Ignore options may be interpreted in the same way, but not changing any part of the code, they are just removing the markers from the suspicious code chunks.

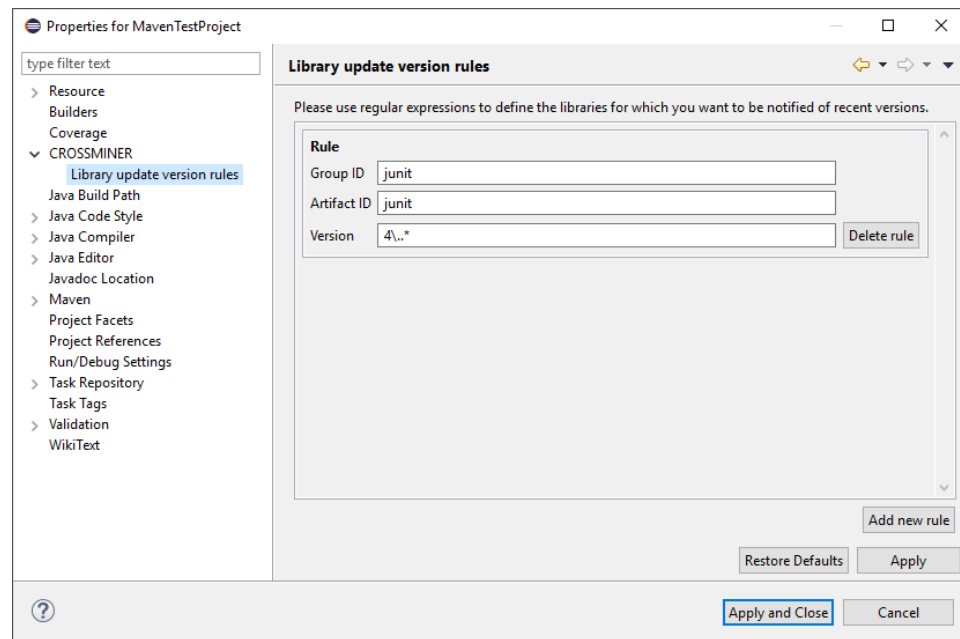


Figure 2.8: Library version rules

2.1.4 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 Code recommendation

Code recommendations are used to retrieve code snippets which may be useful for you by showing an example implementation if a specific library or just presenting a pattern that can be followed. These snippets can be requested by selecting a chunk of code in an open Java source code editor and by clicking on the Request code recommendation option in the context menu's CROSSMINER submenu or by using the Ctrl+Shift+X C key combination. After the Code recommendation view has shown up the suggested patterns

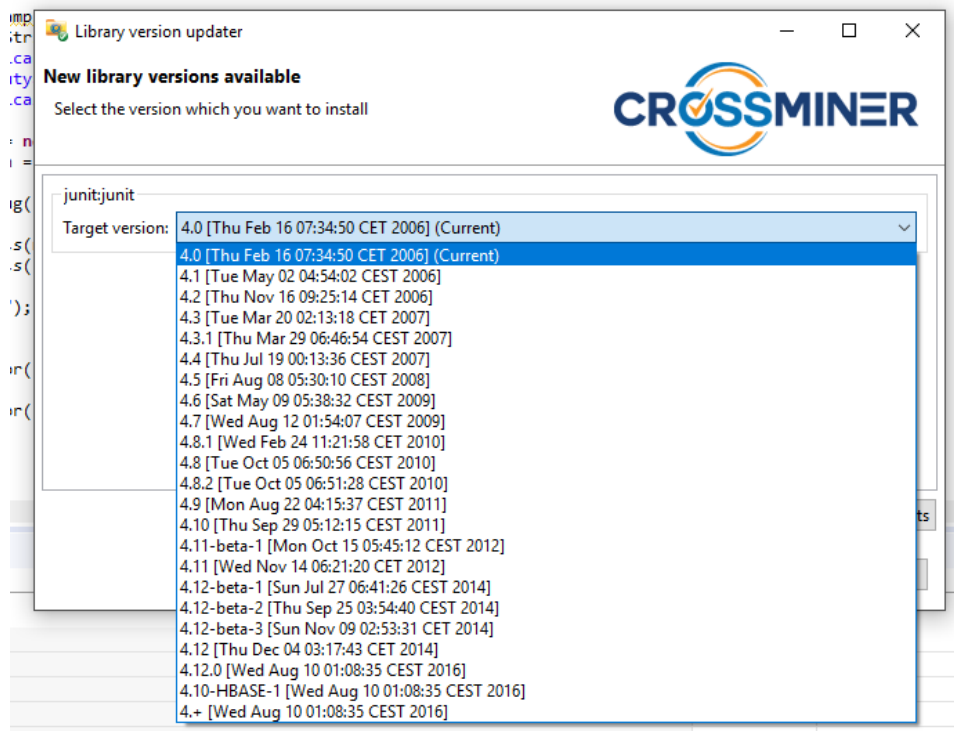


Figure 2.9: Library version Updater - filtered

for this file will be displayed on the left side in a hierarchical representation. The results are grouped by the file, they are relevant to and by the query, which represents the request for the recommendations at a given time using a given part of the file. This can be seen in the Figure 2.11.

By selecting a pattern in the hierarchical tree view the recommended snippet will be shown in the preview view on the right side of the Code recommendation view. Here you can examine the given code, copy a part of it or by using the Insert snippet at cursor button you can paste it into the active editor at the cursor's position or replace the selected code chunk with it. This functionality is available only when there is an active editor opened. The preview view can be seen on the Figure 2.12.

You can also remove the recommended code snippets from the results by opening the context menu over them and choosing the right option for you. Basically there are 4 options that you can go with. If you select a file you have the option to drop all the recommendation that are related to that file. If you select the recommendation query, you can drop all results for that request. If you select the code recommendations you can drop the items one-by-one. Or in all cases there is an option to drop all the recommendations, and this will clear the results view.

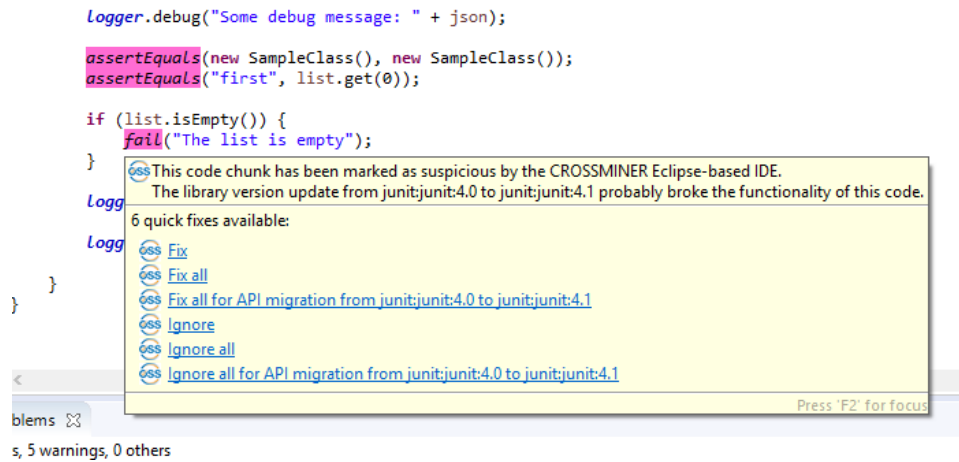


Figure 2.10: Library version Updater - API Migration

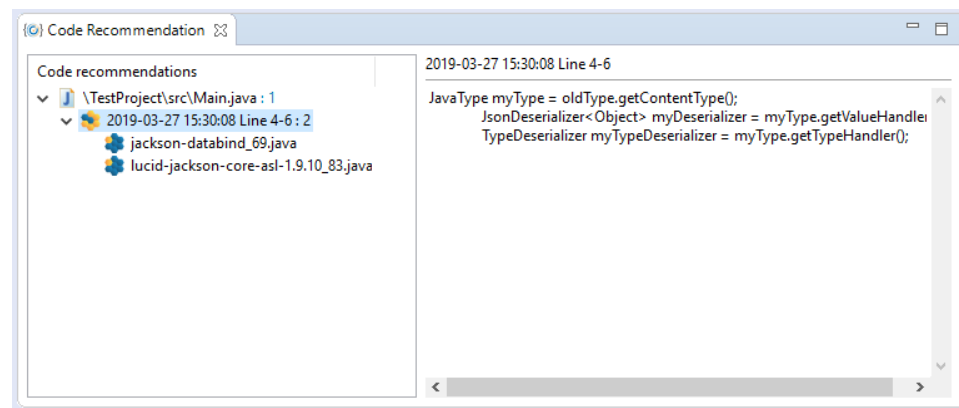


Figure 2.11: Code recommendation view - suggested patterns

By double clicking on a file you can easily navigate to it and request the Eclipse to open it.

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*.

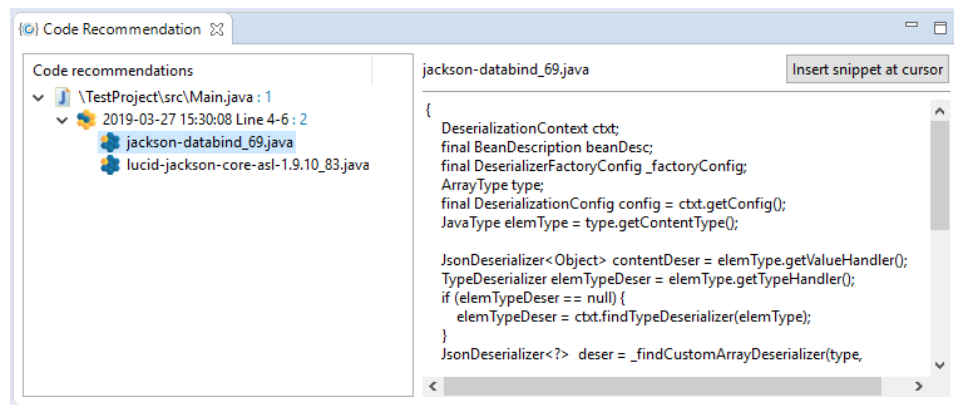


Figure 2.12: Code recommendation view - preview snippet

2.3.1 API documentation and Q&A posts

This feature is used to show the user some documentation or forum posts where she may find some useful information about the API calls that she is using in her code. To initiate a request for these recommendations you have to select a code chunk in an active editor and click on the Request API documentation and Q&A posts options the context menu's CROSSMINER submenu. After this, the results are shown in a particular view, where you can examine these results, copy the link for them or by clicking on the title they can be opened in an external browser. This view can be seen on the Figure 2.13.

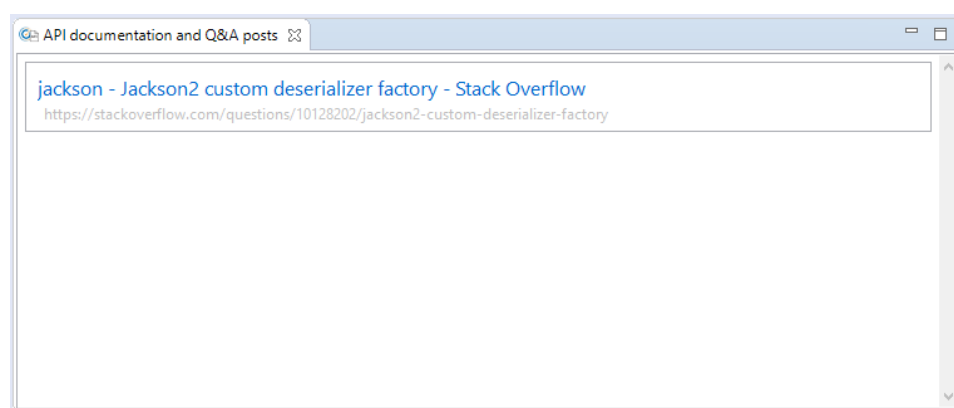


Figure 2.13: API documentation and Q&A posts view - results

2.3.2 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.

2.3.3 Handling API changes

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

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 Database settings

The events are stored in a local database. This database default locate at Workspace location. To change this location you have to open Database preference page under the CROSSMINER tab as figure 3.1. After you change the location it needs an Eclipse restart.

In this page you can also delete your local database. After deletion it also need an Eclipse restart.

3.2 User Activity Monitoring Settings

In the User activity events tab you can customize which events are you want to collect or which metric are you want to compute. As figure 3.2 shows. If you disable an event which is required to calculate a metric, hence that metric won't be computed.

3.3 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.

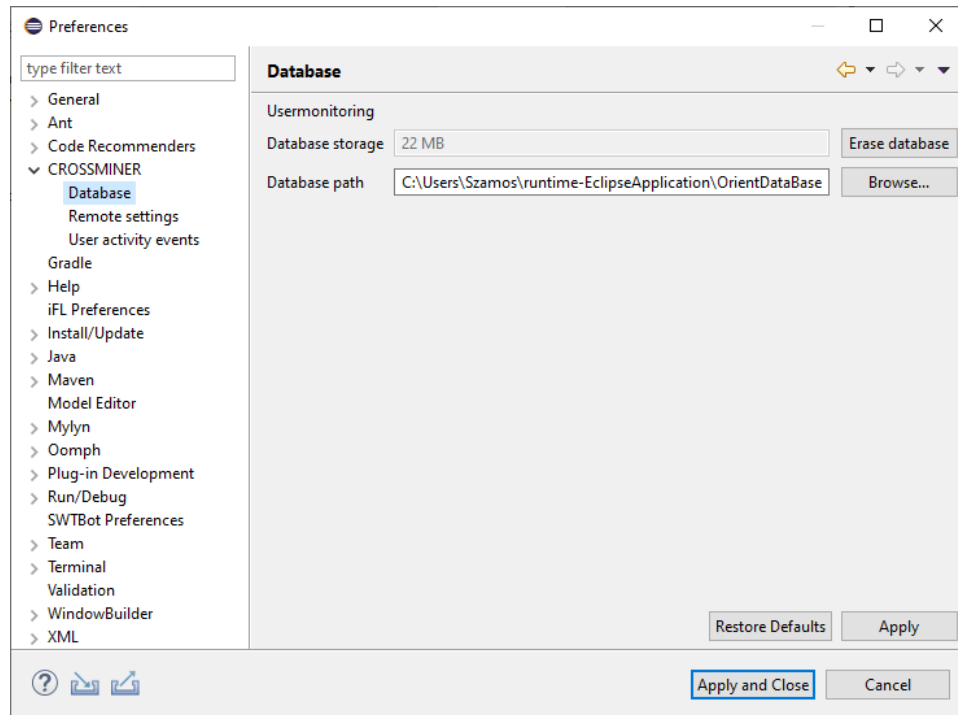


Figure 3.1: Preference page for database settings

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.

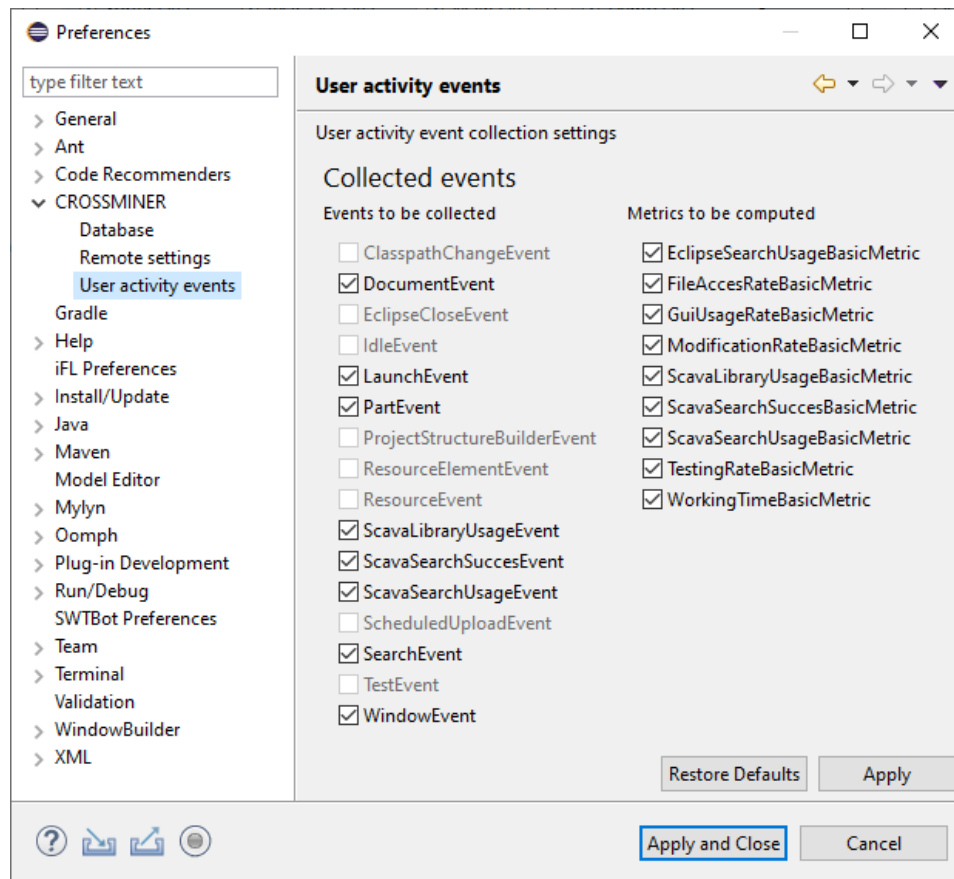


Figure 3.2: Events and metrics settings

3.4 List of Recorded Metrics

3.5 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 .

3.6 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.

3.7 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 Postman¹

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

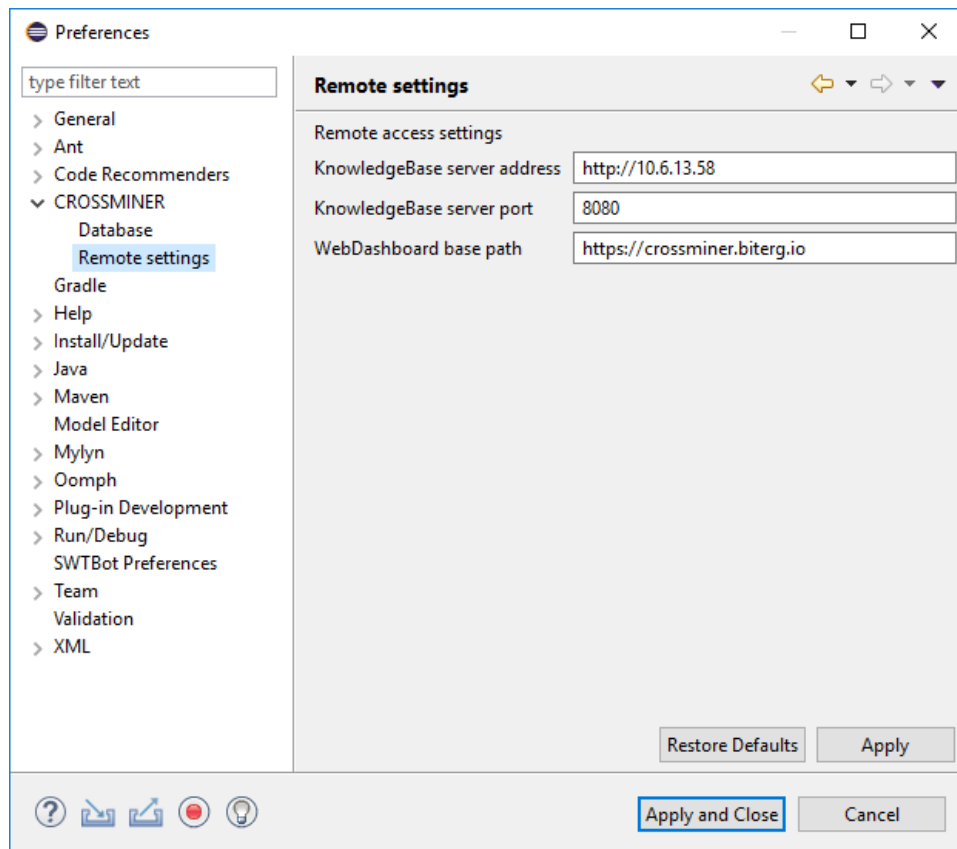


Figure 4.1: Plug-in remote settings