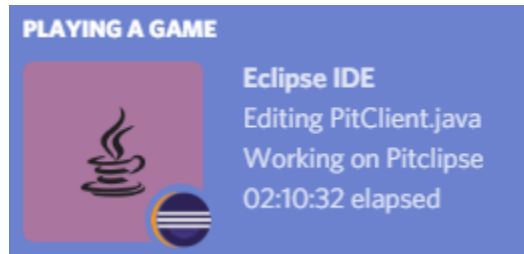

Discord Rich Presence for Eclipse IDE Documentation

Emmanuel CHEBBI

May 01, 2020

1	Table of Contents	3
1.1	From the Eclipse Marketplace	3
1.2	From the P2 repository	3
1.3	Differentiate global and project scopes	4
1.4	Change displayed project names	6
1.5	Change wording	7
1.6	Change language icons	9
1.7	Hide some information	12
1.8	Deactivate the plug-in	13
1.9	Support new editors	14
1.10	Change wordings	17
1.11	Contributing Guide	18
1.12	Discord does not detect Eclipse IDE as a game	18
1.13	A ‘Workbench early startup error’ occurs since the plug-in is installed	18

The *Discord Rich Presence for Eclipse IDE* is a plug-in for Eclipse IDE that shows on Discord the projects and files you're currently working on:



This plug-in can be installed on any application and IDE based on Eclipse Mars 1 or newer. Please see the following table of contents for instructions on how to install the plug-in, change its preferences or extend it.

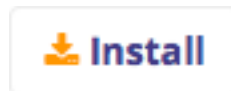
Tip: See also the sources on [GitHub](#)!

1.1 From the Eclipse Marketplace

The plug-in is available on the Eclipse Marketplace:

- <https://marketplace.eclipse.org/content/discord-rich-presence-eclipse-ide>

You can also drag the following button and drop it into your workspace to start the installation:



1.2 From the P2 repository

All versions of the plug-in are also deployed on the following P2 repository:

- <https://dl.bintray.com/kazejiyu/eclipse-discord-integration/updates/>

To use it:

1. Open Eclipse IDE
2. Click on `Help > Install new software...`
3. Paste the P2's URL in the `Work with` field

Under the `Discord Rich Presence Integration` category, check the following features:

- `Discord Rich Presence`
- `Discord Rich Presence UI`

Note: The latter is not required for the plug-in to work but adds dedicated preferences pages to the IDE Preferences and Project Settings dialogs.

Click on `Next >`, accept the licenses then `Finish`.

Wait for the installation to end then restart the IDE.

1.3 Differentiate global and project scopes

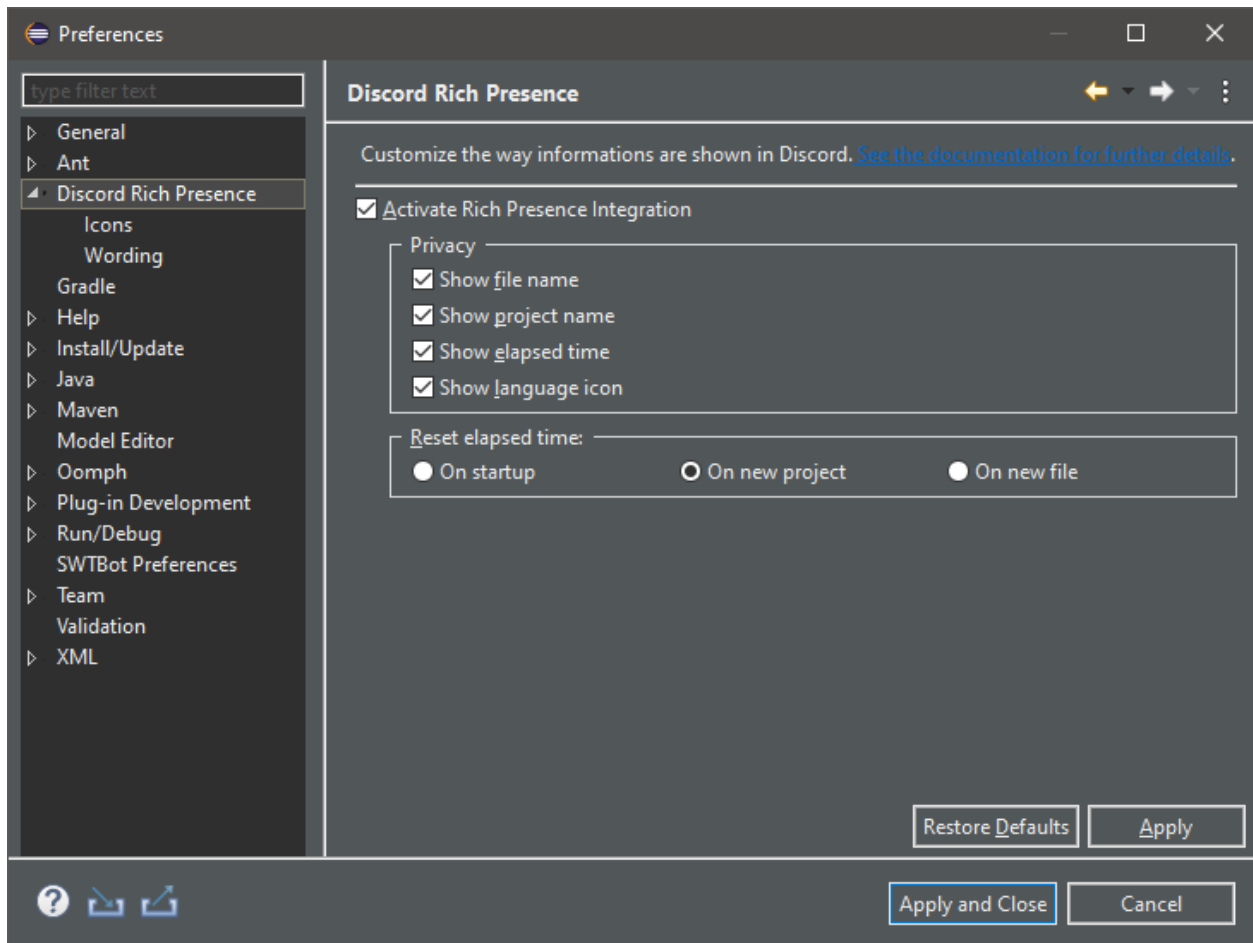
The *Discord Rich Presence for Eclipse IDE* plug-in offers numerous preferences allowing you to choose the information to show on Discord. Most of these preferences can be changed at both global and project scope.

1.3.1 Global scope

Preferences changed at the global scope are applied to all the projects in the active workspace. They are not shared between workspaces so different workspaces can have different global preferences.

Those preferences can be edited through the *Preferences* dialog:

1. Go to `Window > Preferences`
2. Click on `Discord Rich Presence`
3. The following page should open:



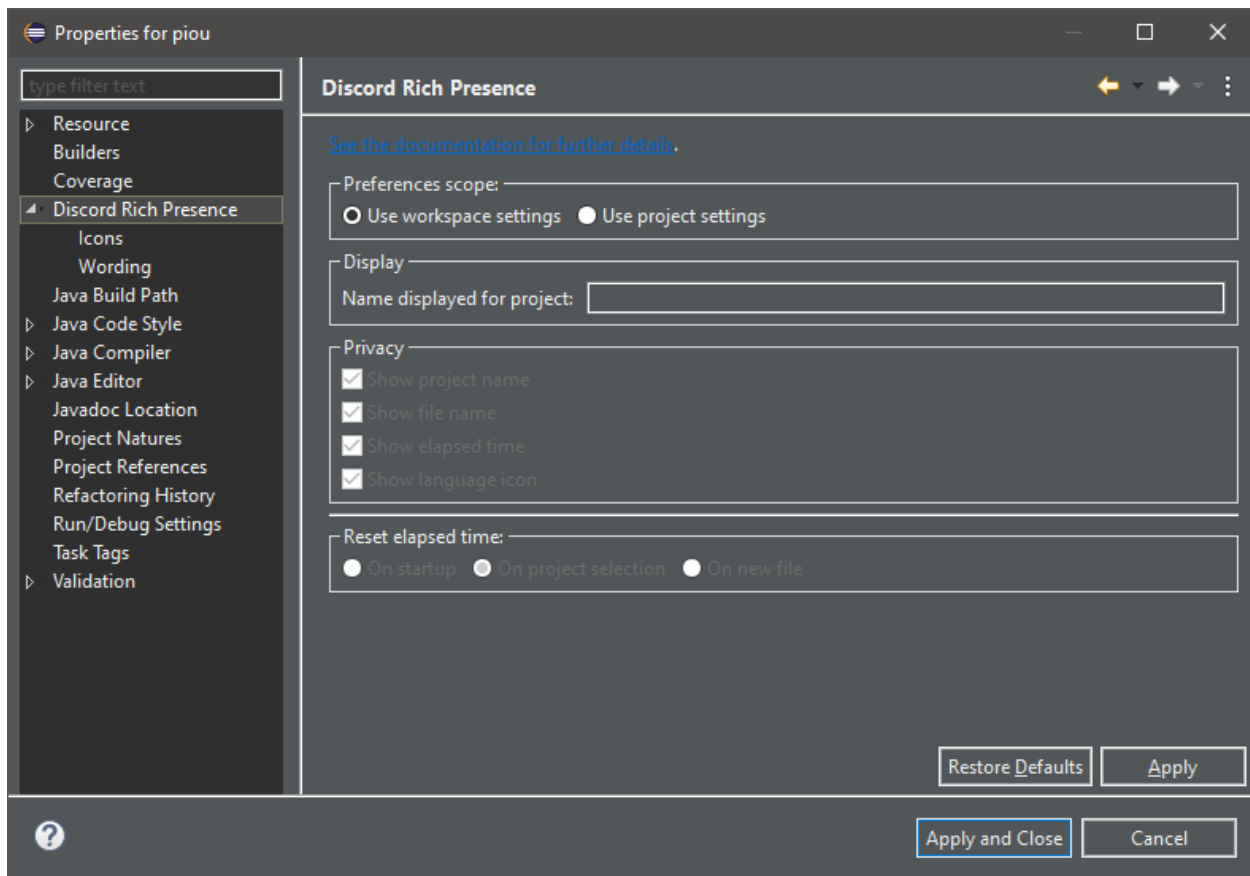
See the different sections on the left menu to discover what can be customized.

1.3.2 Project scope

Preferences changed at the project scope are only applied to one project. They have precedence over global scope.

Those preferences can be edited through the *Project Settings* dialog:

1. Right-click on a project > Properties
2. Click on Discord Rich Presence
3. The following page should open:



By default each project uses the global preferences (notice the `Use workspace settings` checkbox). In order to set the preferences at the project-scope you have to check the `Use project settings` checkbox.

1.4 Change displayed project names

By default the plug-in shows on Discord the name of the Eclipse project. However, it may happen that this name is not really accurate. This may happen, for example, when the different Eclipse projects are sub-modules of a bigger one. In such a case you may want to provide a custom name that will be used instead.

The name shown for a given Eclipse project can be changed in the project's preferences page:

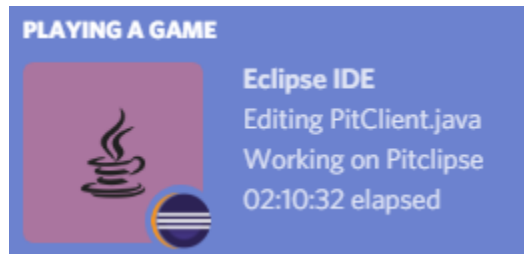
1. Right-click on the project > Properties
2. Click on `Discord Rich Presence`
3. Look at the `Display` section
4. Type a name in `Name displayed for the project` text field
5. Click on `Apply`

Note: This preference is the only one that is taken into account even if the `Use project settings` preference is unchecked. This allows you to customize a project's name while benefiting from global preferences.

1.5 Change wording

1.5.1 Motivation

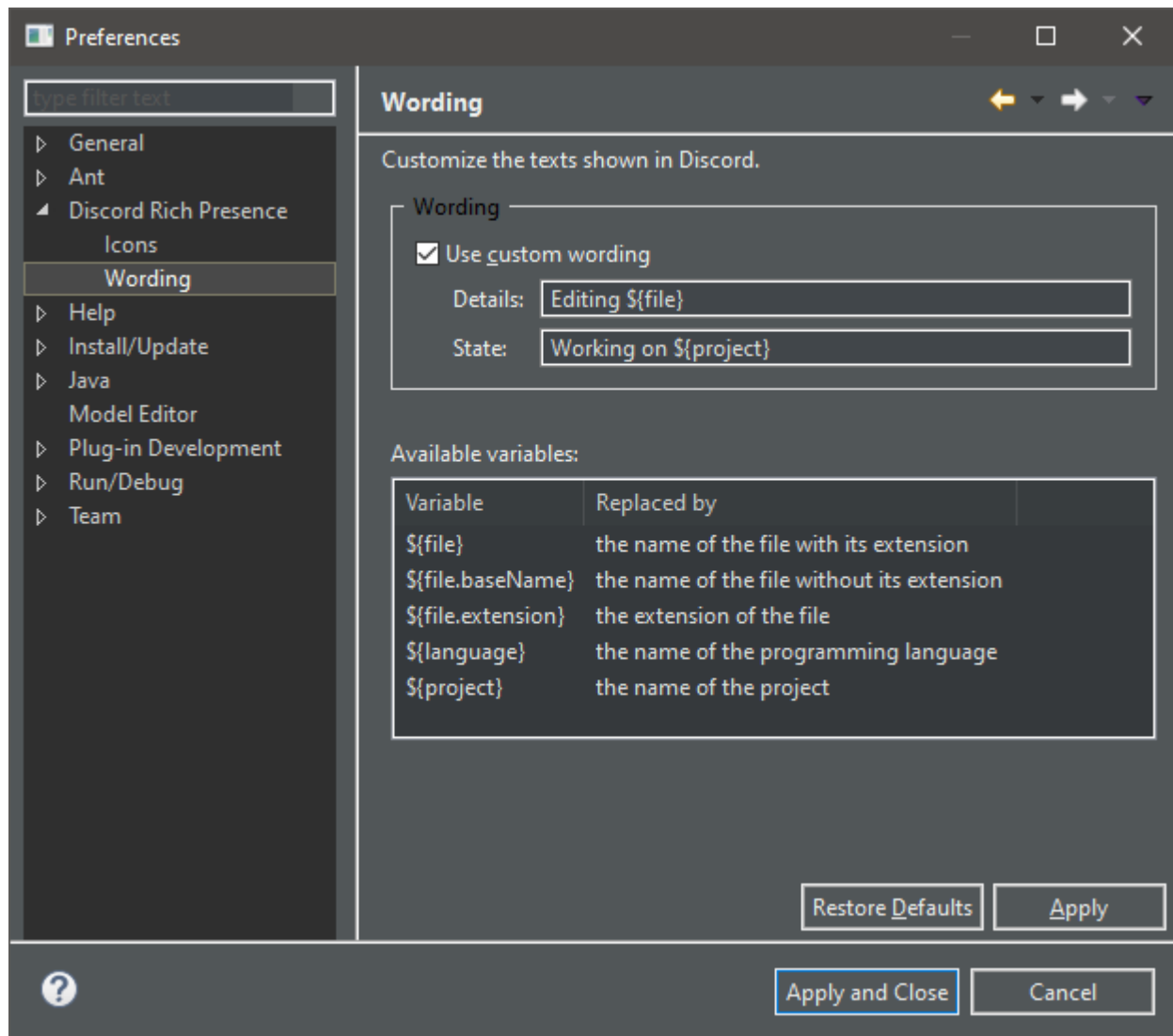
By default, the plug-in uses the following sentences: *Editing <file name>* and *Working on <project>*:



1.5.2 Provide custom templates

However, they may not fit your needs. Hopefully you can specify your own sentences from the preferences page:

1. Click on `Window > Preference`
2. Select `Discord Rich Presence > Wording`
3. The following page should open:



4. Check `Use custom wording`
5. Fill the *Details* and *State* fields with your own templates

Tip: In Discord, *Details* is the top text and *State* the bottom text. See also [Discord's Rich Presence documentation](#).

The following variables can be used in the templates:

Variable	Replaced by
<code>\${file}</code>	the name of the file with its extension
<code>\${file.baseName}</code>	the name of the file without its extension
<code>\${file.extension}</code>	the extension of the file
<code>\${language}</code>	the name of the programming language
<code>\${project}</code>	the name of the project

The wording can also be customize per project:

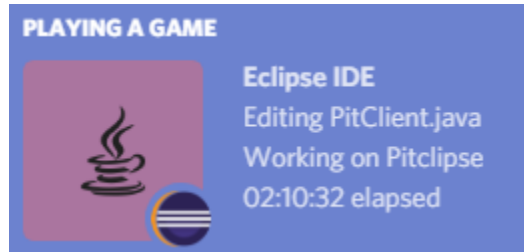
1. Right-click on a project > `Properties`

2. Select `Discord Rich Presence > Wording`

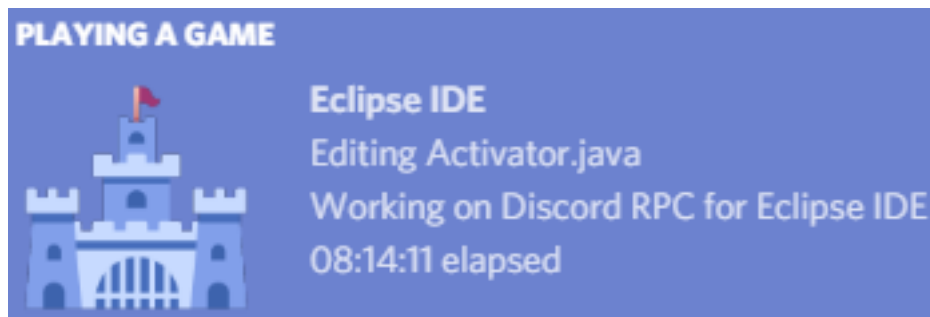
1.6 Change language icons

1.6.1 Motivation

By default, the icon displayed on Discord is a flat icon of the programming language used in the active editor. Since the programming language is usually determined upon the edited file's extension, editing a `*.java` file will show a Java icon on Discord:



However, you may want to use custom icons, because the default ones don't suit you or because you'd like the icon to represent the project you're working on instead of the programming language you're using:



Credit

Icon by [Chanut is Industries](#)

Published on [IconFinder](#)

Under the [Creative Commons \(Attribution 3.0 Unported\)](#) licence

1.6.2 Limitations due to Discord API

Ideally, you would just have to provide the plug-in a set of icons and *voilà*, you would be done. However, at the moment Discord does not provide any API allowing developers to programmatically upload new icons. To overcome this limitation, here is what you have to do:

1. Create a Discord application
2. Upload your custom icons on this application
3. Ask the *Discord Rich Presence for Eclipse IDE* plug-in to use your own Discord application

The following section explains step by step how to proceed.

1.6.3 Create a Discord application

1. Connect to the [Discord Developer Portal](#). You'll have to login with your Discord credentials.
2. Click on the `New Application` button, on the top-right of the screen.
3. Type the name of your application. It will be shown on Discord instead of *Eclipse IDE* (which is the name of the default Discord application used by the plug-in).
4. Click on `Create`.
5. The main page of the application should open. Click on the *Rich Presence* item, which is in the menu navigation on the left.
6. Click on `Add Image(s) in the RICH PRESENCE ASSETS` section.
7. A dialog should open. Select the icons you want to use then validate.
8. A list of all the icons you selected should be displayed below. In order to register the icons, you still have to give each of them an ID. This ID is used by the Eclipse plug-in to select the right icon for the right language. See the table below for a list of available IDs.
9. Click on `Save Changes`.

ID	File patterns
ada	*.ada
binary	*.bin
boo	*.boo
clojure	*.clj
cobol	*.cob, *.cbl, *.cpy
c	*.c, *.h
cpp	*.cpp, *.hpp
crystal	*.cr
css3	*.css
csharp	*.cs
dart	*.dart
docker	Dockerfile
fortan	*.f, *.for, *.f90, *.f95, *.f03
git	*.git, .gitignore, .gitattributes, .gitmodules
go	*.go
gradle	*.gradle
groovy	*.gvy, *.gy, *.gsh
haskell	*.hs
html5	*.html
java	*.java
js	*.js
kotlin	*.kt, *.ktm, *.kts
bibtex	*.tex
lisp	*.lisp, *.lsp
lua	*.lua
markdown	*.markdown, *.mdown, *.md
ocaml	*.ml, *.mli
pascal	*.pas

Continued on next page

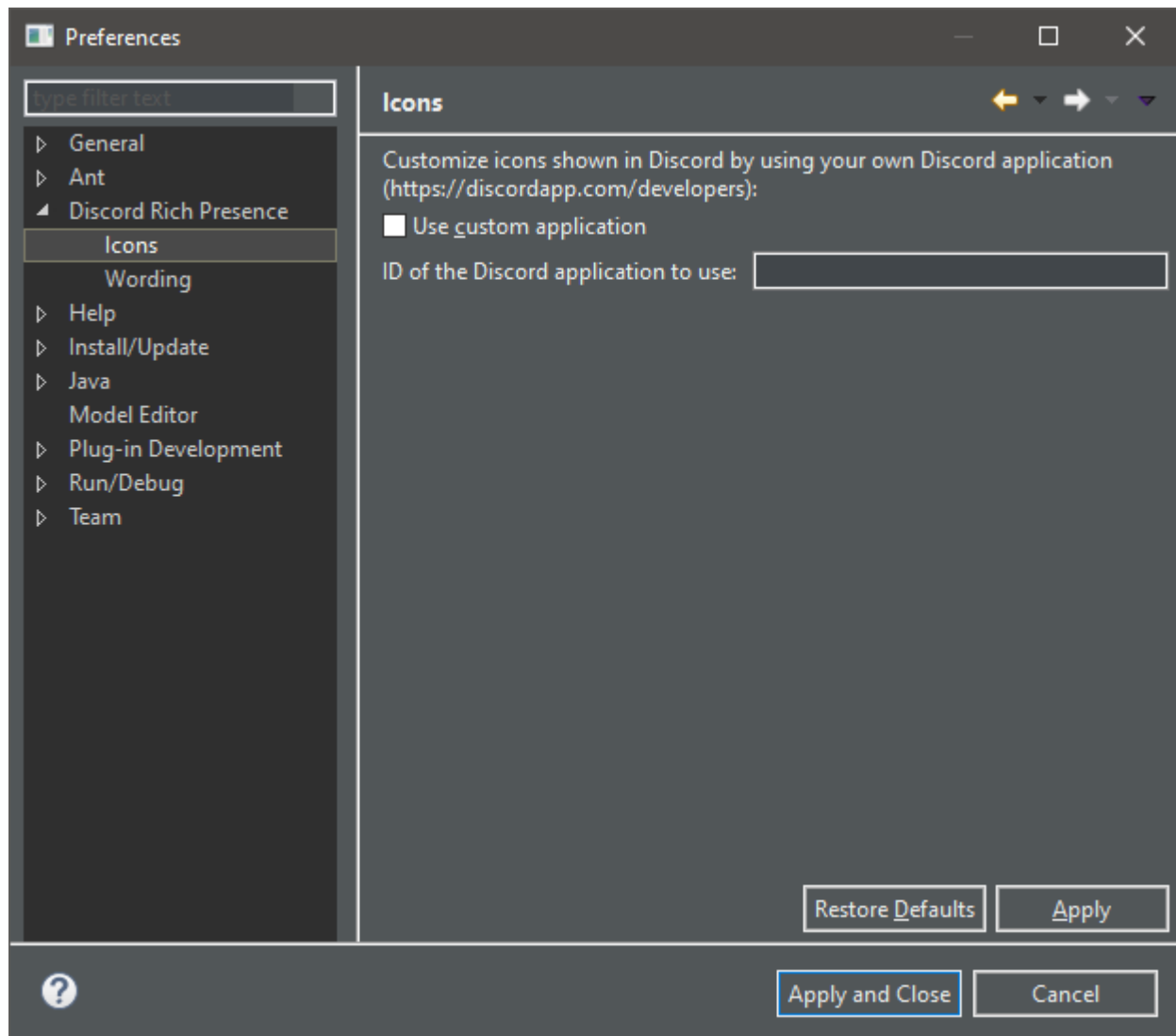
Table 1 – continued from previous page

ID	File patterns
php	*.php
prolog	*.pro, *.pl
python	*.py
r	*.r
ruby	*.rb
rust	*.rs, *.rlib
scala	*.scala, *.sbt
sql	*.sql
swift	*.swift
terminal	*.sh, *.bash, *.ksh
text	*.txt
ts	*.ts

1.6.4 Configure the plug-in to use your application

Once your Discord application is ready you have to ask the Eclipse plug-in to use it:

1. Open Eclipse IDE
2. Click on `Windows > Preferences`
3. Select `Discord Rich Presence > Icons`
4. The following page should open:



5. Check `Use custom application`
6. Specify the ID of your application (it's the `Client ID` field in your Discord application's main page)
7. Click on `Apply`

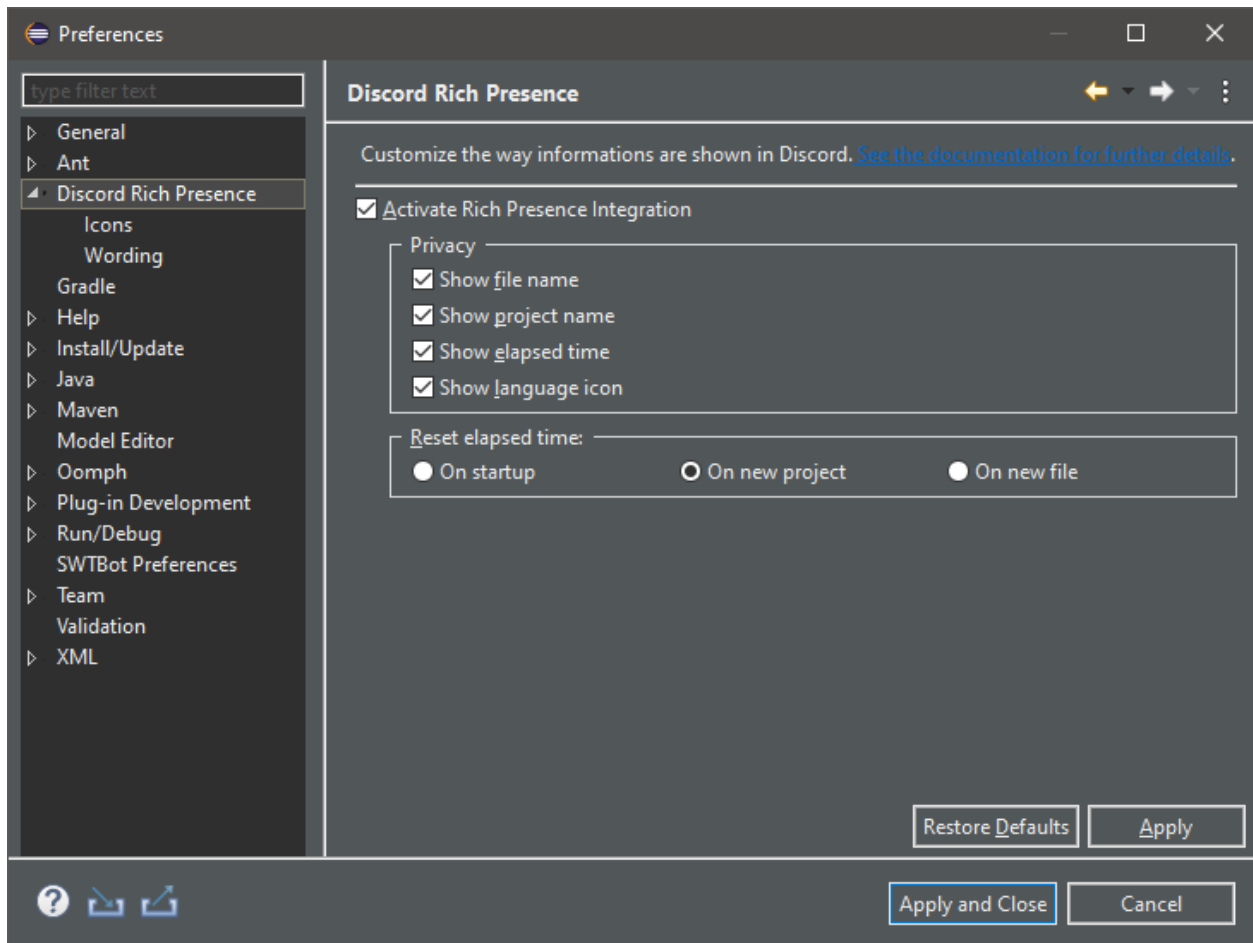
Discord should be updated with your own icons!

Tip: A custom Discord application can also be specified *per project*.

1.7 Hide some information

Plug-in's preferences allow you to prevent some information to be shown on Discord. These preferences can be set at both *global and project scopes*.

Open the preferences page and take a look at the *Privacy* section:



The checkboxes allow you to select whether the following information should be shown on Discord:

- the name of the edited file,
- the name of the edited file's project,
- the time elapsed working,
- the icon of the edited file's programming language.

1.8 Deactivate the plug-in

Once the plug-in is installed, Discord Rich Presence is automatically started on Eclipse startup. As a result, Discord displays **Playing Eclipse IDE** as a status message as soon as Eclipse IDE is opened. This behavior may not be desirable but can be easily deactivated.

1.8.1 Deactivate occasionally

The plug-in provides a preference to deactivate its behavior:

1. Open Eclipse Preferences (Window > Preferences)
2. Open the plug-in's page (Discord Rich Presence)
3. Uncheck *Activate Rich Presence Integration*

4. Click on *Apply*

From that time on, Discord won't be notified anymore by Eclipse IDE. In order to re-activate Rich Presence, follow the steps above and check *Activate Rich Presence Integration* again.

1.8.2 Deactive the plug-in for a long time

While convenient, the previous method is not really optimized. Indeed, even if no Rich Presence is shown on Discord the plug-in is still activated and is notified each time a new editor opens, which may cause a little overhead.

Follow the steps below to completely deactivate the plug-in:

1. Open Eclipse Preferences (Window > Preferences)
2. Open the *Startup and Shutdown* page (General > Startup and Shutdown)
3. Uncheck *Discord Rich Presence for Eclipse IDE*
4. Reboot Eclipse IDE

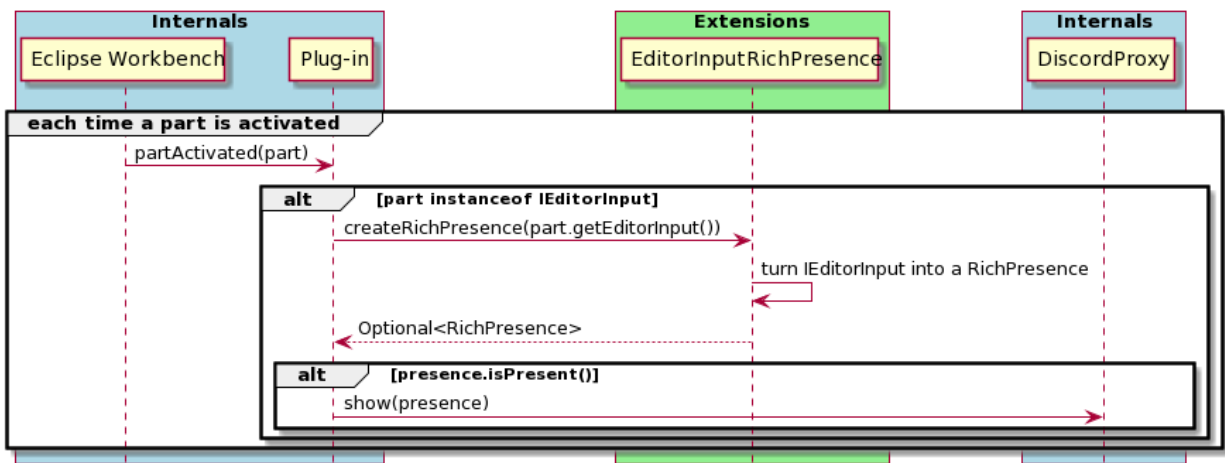
From that time on, Discord won't be notified anymore by Eclipse IDE. In order to re-activate Rich Presence, follow the steps above and check *Discord Rich Presence for Eclipse IDE* again then reboot the IDE.

1.9 Support new editors

Important: This page requires some knowledge about [Eclipse RCP development](#) and [Eclipse Extension Points](#).

1.9.1 How editors are handled by the plug-in

The *Discord Rich Presence for Eclipse IDE* plug-in shows on Discord information extracted from the currently active editor. In order to gather these information, it proceeds as depicted in the following sequence diagram:



Basically:

1. The plug-in is notified each time a new `part` is opened.
2. If the `part` is an instance of `IEditorPart` then we get its `IEditorInput` (which represents the underlying element being edited).

3. This `IEditorInput` is broadcasted to an instance of `EditorInputRichPresence`, an interface defined by this plug-in.
4. The `EditorInputRichPresence` instance turns the `IEditorInput` into a `RichPresence`.
5. The `RichPresence` is then shown on Discord through a proxy.

Most of this stuff is managed internally so you don't have to bother with it. The only thing you'll have to do is to register a new `EditorInputRichPresence` (see the [Adapt a new `IEditorInput`](#) section below).

1.9.2 Editors supported out of the box

By default, almost all text-based editors should be taken into account. More specifically, all the editors which input is an instance of one of the following classes are handled:

- `IFileEditorInput`,
- `FileStoreEditorInput`.

1.9.3 Adapt a new `IEditorInput`

Provide a dedicated extension

You can provide an adapter for a given `IEditorInput` by contributing to the `fr.kazejiyu.discord.rpc.integration.editor_input_adapter` extension point. You'll have to provide a class that implements the `EditorInputRichPresence` interface.

This interface is defined as follows:

```

1  /**
2   * Extracts a RichPresence from an IEditorInput.
3   *
4   * This interface should be implemented by clients who aim to define
5   * the information shown in Discord for their own editor.
6   */
7  public interface EditorInputRichPresence extends Comparable<EditorInputRichPresence> {
8
9      /**
10       * Helps to choose an adapter over another when several ones
11       * are registered for the same {@code IEditorInput}.
12       *
13       * The higher the priority, the more the adapter will be favored.
14       *
15       * For instance, given two adapters registering themselves for inputs of type
16       * FileEditorInput and which priorities are 0 and 1, then the adapter
17       * of priority 1 will be chosen to handle the input.
18       *
19       * Built-in adapters have a priority of 0. Hence, giving a higher priority
20       * ensures that the adapter will be preferred over default ones. This allows
21       * to dynamically override other adapters if needed.
22       *
23       * It is advised to only choose tens, such as 10 or 20, instead of digits
24       * so that it is easier to add new adapters later if needed.
25       *
26       * @return the priority associated with this adapter.
27       */
28      int getPriority();

```

(continues on next page)

(continued from previous page)

```

29
30     /**
31      * Returns the class of the input expected as an argument of
32      ↪ createRichPresence(GlobalPreferences, IEditorInput)}.
33      * @return the class of the input expected as an argument of
34      ↪ createRichPresence(GlobalPreferences, IEditorInput)}
35      */
36     Class<? extends IEditorInput> getExpectedEditorInputClass();
37
38     /**
39      * Creates the Rich Presence information to send to Discord.
40      *
41      * Important: this method may be called several times in a row with the same
42      ↪ editor input.
43      *
44      * @param preferences
45      *         User's preferences regarding the information to show in Discord.
46      *         Must not be null.
47      * @param input
48      *         The input of the active editor.
49      *         Must satisfy getExpectedEditorInputClass().isInstance(input) ==
50      ↪ true.
51      *
52      * @return the information to show in Discord if the input can be handled
53      */
54     Optional<RichPresence> createRichPresence(GlobalPreferences preferences,
55      ↪ IEditorInput input);
56 }

```

As you can see, you have to implement 3 different methods. Their use is detailed in the JavaDoc, but I believe that the third one may benefit from additional hints.

Manage user's preferences

First of all, the `createRichPresence` is responsible of actually creating a Rich Presence that will be shown on Discord from an editor's input. Its first parameter, `preferences`, is here because it is your responsibility to ensure that the information shown in Discord follow user's preferences. The `GlobalPreferences` class represents the preferences set by the user at *global scope*. If you manage to find the `IProject` associated to the `IEditorInput` you use the following snippet to get the applicable preferences (either project or global scope):

```

UserPreferences applicablePreferences = preferences.
↪ getApplicablePreferencesFor(project);

```

Create a new RichPresence

The `ImmutableRichPresence` class allows you to create a new Rich Presence easily. See the [official RPC documentation](#) for a description of how available fields are shown on Discord.

Hint: If no Rich Presence can be created from the given `IEditorInput` (for instance in the case where the input does not provide enough information) then you should not throw any exception but rather return `Optional.empty()` instead.

Example

The following snippet is extracted from `DefaultFileEditorInputRichPresence`, a built-in adapter, and shows how a `RichPresence` is created from an editor input:

```

1 @Override
2 public Optional<RichPresence> createRichPresence(GlobalPreferences preferences,
3     ↪ IEditorInput input) {
4     if (!(input instanceof IFileEditorInput)) {
5         throw new IllegalArgumentException("input must be an instance of " +
6     ↪ IFileEditorInput.class);
7     }
8     IFileEditorInput fileInput = (IFileEditorInput) input;
9     IFile file = fileInput.getFile();
10    IProject project = file.getProject();
11
12    UserPreferences applicablePreferences = preferences.
13    ↪ getApplicablePreferencesFor(project);
14
15    ImmutableRichPresence presence = new ImmutableRichPresence()
16        .withProject(project)
17        .withDetails(detailsOf(applicablePreferences, file))
18        .withState(stateOf(applicablePreferences, project))
19        .withLanguage(languageOf(applicablePreferences, file))
20        .withLargeImageText(largeImageTextOf(applicablePreferences, file));
21
22    return Optional.of(presence);
23 }

```

1.10 Change wordings

Important: This section requires some knowledge about [Eclipse RCP development](#) and [Eclipse Extension Points](#).

You must also have read [Support new editors](#).

1.10.1 Motivation

By default, information is displayed on Discord using the following format:

It is very simple but maybe that it does not fulfil your needs. Read below to learn how to change it.

1.10.2 Implementation

I plan to add preferences allowing one to customize these default wordings easily from the Eclipse IDE (see [issue #71](#)) but it will take some time so in the meantime you'll have to get your hands a little dirty.

In order to provide your own wording you have to create two `EditorInputToRichPresenceAdapter` as explained in [Support new editors](#).

These adapters must:

- expect editor input which are respectively instances of `IFileEditorInput` and `FileStoreEditorInput`,

- have a priority greater than 0 in order to be favoured over built-in ones.

You can now implement the `createRichPresence` method as you wish to create a Rich Presence that follows your own wording.

Tip: I advise you to create those adapters by copying the code of:

- `DefaultFileEditorInputRichPresence`
 - `DefaultURIEditorInputRichPresence`
-

1.11 Contributing Guide

Please read the [contributing guide](#) for a technical introduction to the project.

1.12 Discord does not detect Eclipse IDE as a game

If Discord is not showing anything, ensure that Eclipse IDE is detected as a game. To this end:

1. Open *Discord User Settings*
2. Go to the *Games* tab
3. Click on the *Add it!* button
4. Select *Eclipse IDE*.

1.13 A ‘Workbench early startup error’ occurs since the plug-in is installed

In case you encounter the following error on startup:

you can try to modify the `eclipse.ini` file which is located next to `eclipse.exe` (within Eclipse IDE’s installation directory) in order to set the `jna.nosys` property to true as follows:

Note: The `-vmargs` line should already exist, otherwise you can append it at the end of the file.
