



12 - Eclipse Builders, Natures and Markers:

Page 2 of 3

[↑ TOC](#)[◀ 11 - Eclipse Plugin Editors Tutorial](#)[13 - Eclipse Plugin Help Tutorial ▶](#)

Markers

As described earlier, Markers are used mark locations in editor. Java editor uses these markers to mark error locations, warnings, book marks etc.

There are many types of predefined markers defined as constants in IMarker interface.

`org.eclipse.core.resources.bookmarkIMarker.BOOKMARK`

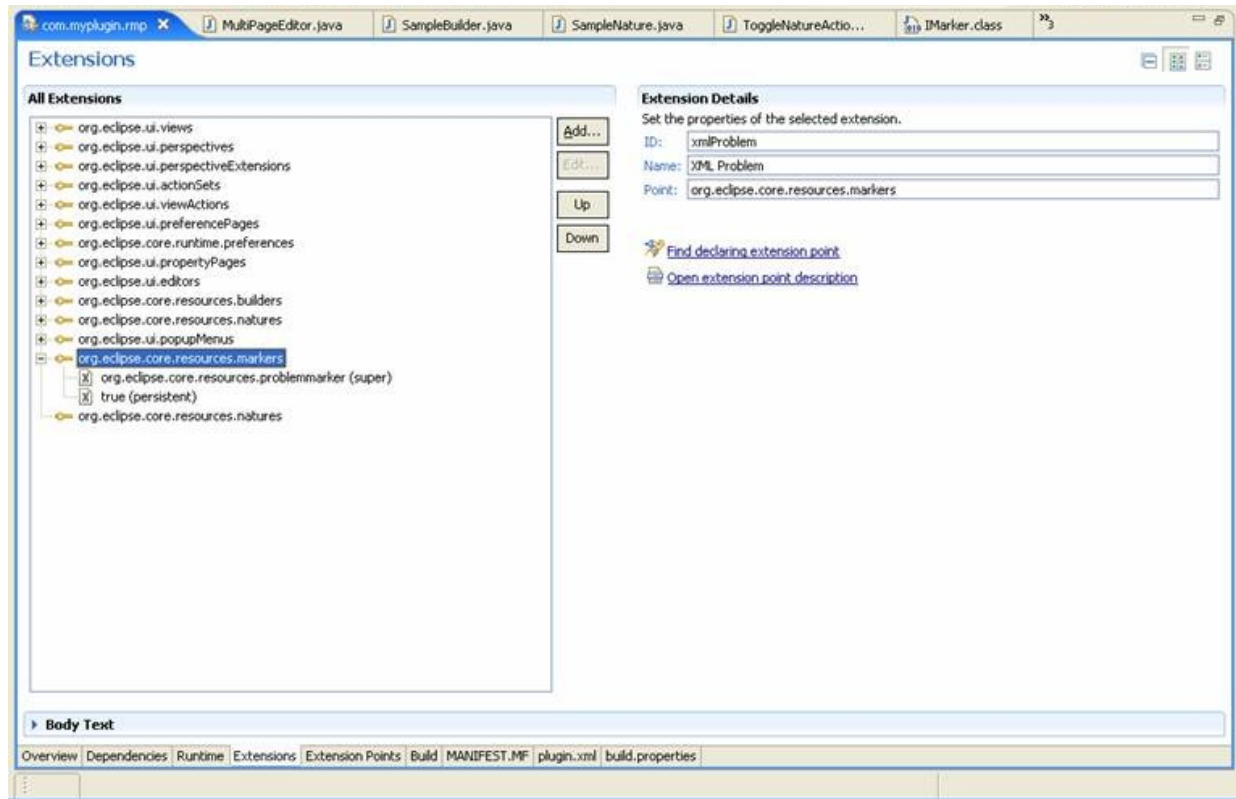
`org.eclipse.core.resources.markerIMarker.MARKER`

`org.eclipse.core.resources.problemmarker IMarker.PROBLEM`

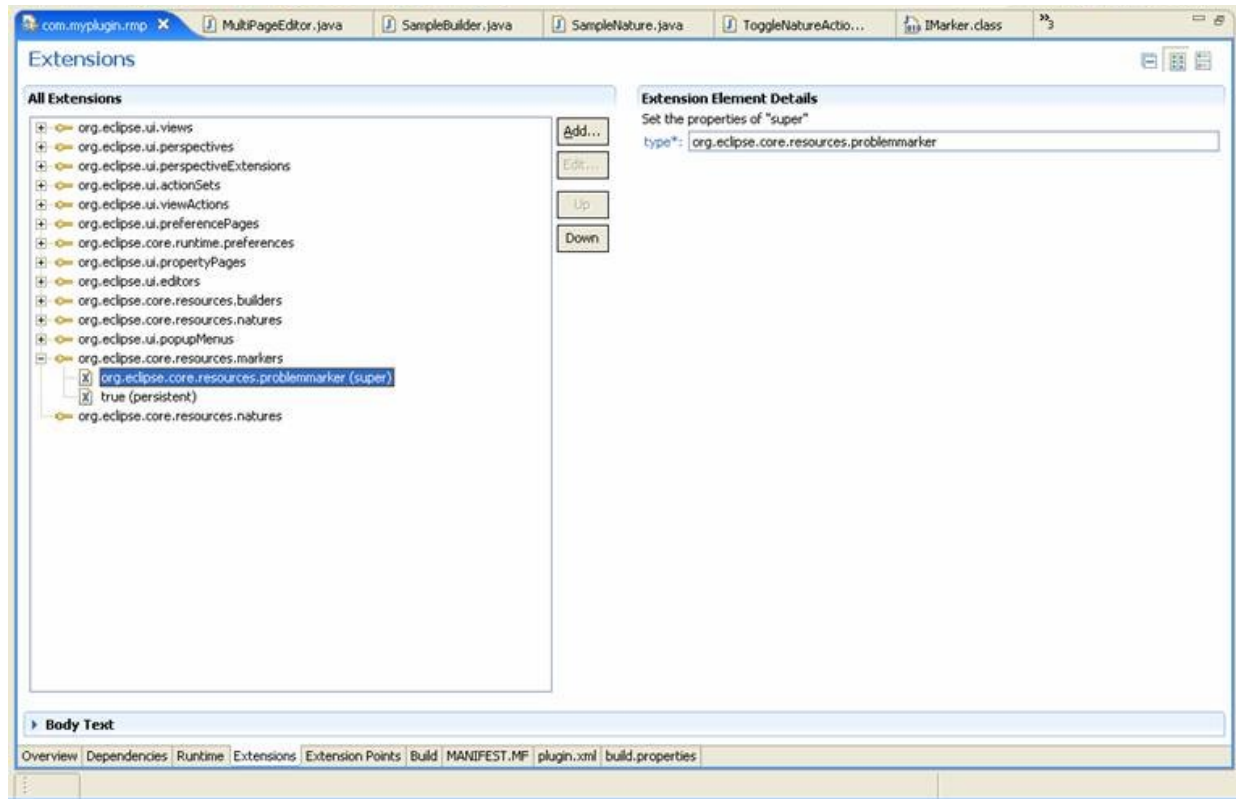
`org.eclipse.core.resources.taskmarkerIMarker.TASK`

`org.eclipse.core.resources.textmarkerIMarker.TEXT`

Let's review the plug-in manifest file to see what it takes to create a new marker. New marker is created by extending `org.eclipse.core.resources.markers` extension point. ID and Name are data value and name respectively. Point is nothing but the extension point we are using to create this marker.



Next, we need to define the super type marker for new marker. Since we want to show errors in Problems View we are using `org.eclipse.core.resources.problemmarker` as supertype.



Next is the persistent element. Since we want this markers to persist across eclipse

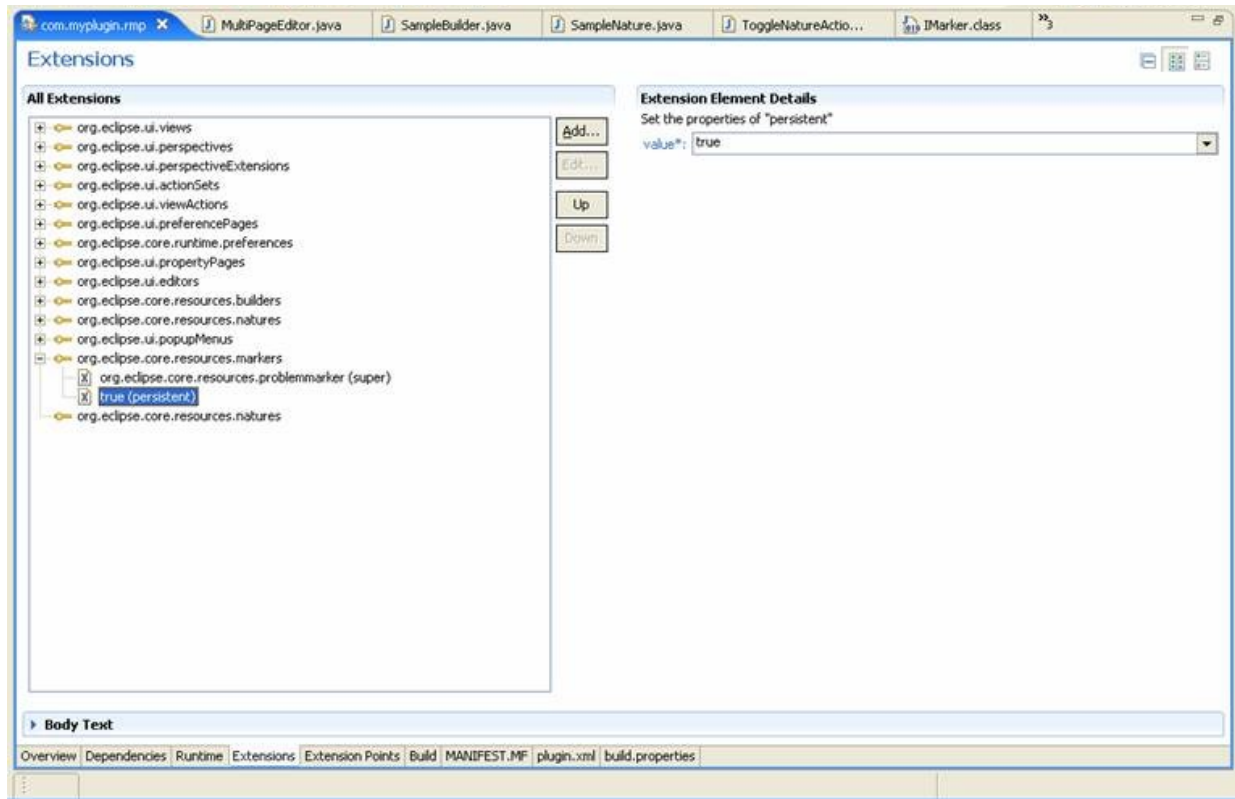


S

[Technology](#)[Eclipse Plugin Tutorial](#)

12 - Eclipse Builders, Natures and Markers

[Login](#)



Reviewing the generated code

addMarker method in SampleBuilder class is used to show above defined marker in the editor whenever problem/error occurs.

```

1  private static final String MARKER_TYPE = "com.myplugin.rmp.xml!";
2  private void addMarker(IFile file, String message, int lineNumber) {
3      try {
4          IMarker marker = file.createMarker(MARKER_TYPE);
5          marker.setAttribute(IMarker.MESSAGE, message);
6          marker.setAttribute(IMarker.SEVERITY, SEVERITY_ERROR);
7          if (lineNumber == -1) {
8              lineNumber = 1;
9          }
10         marker.setAttribute(IMarker.LINE_NUMBER, lineNumber);
11     } catch (CoreException e) {
12     }
13 }

```

IMarker is the main interface when we are dealing with eclipse markers. Please refer to [Eclipse API Specification](#) to know more about IMarker Interface.

[↑ TOC](#)[← 11 - Eclipse Plugin Editors Tutorial](#)[13 - Eclipse Plugin Help Tutorial →](#)

Like us on Facebook



Wideskills
798 likes

Like Page

Be the first of your friends to like this

