

DeMark User Manual

Team DeFacto-UW:

Andrew Tran, Tony Vo, Tuan Ma, Jeff Xu, Lemei Zhang

April 19, 2018

1 Introduction

Debugging plays an important role in software engineering, and it can be very time-consuming. One common way for a developer to debug through their code is by adding temporary, non-production lines of code, such as print statements, to understand the program's state, but it can be a tedious task to manage them manually.

DeMark is a plugin for the IntelliJ IDEA that is used to keep track of temporary debug code. It aims to increase software developers' programming productivity by facilitating the management of temporary code.

2 Installation

The following instruction is a modified version of the official IntelliJ help documentation on the JetBrains website^[1]:

1. Press **[Ctrl + Alt + S]** or choose **File | Settings** (for Windows and Linux or **IntelliJ IDEA | Preferences** (for macOS) from the main menu, then go to **Plugins**.
2. Click the **Install JetBrains plugin** or the **Browse repositories** button.
3. In the dialog that opens, search for the plugin by typing "DeMark" into the search bar. Right-click the plugin labeled "DeMark" and select **Download and Install**.
4. Confirm your intention to download and install DeMark.
5. Close the plugin window and click **OK** in the **Settings** dialog.
6. Restart IntelliJ IDEA.

DeMark will appear as a drop down menu on the top tool bar. There will also be a toggle button next to the quick tools area.

3 Using DeMark

For each behavior, the keyboard shortcuts listed are their default shortcuts, the user can change the key bindings to any functions of DeMark if they choose to. To configure keyboard shortcuts for IntelliJ, please follow the instructions on this page: <https://www.jetbrains.com/help/idea/configuring-keyboard-and-mouse-shortcuts.html>.

3.1 Mark and Unmark

Marking a line will highlight the line and put a check mark next to its line number. There are two ways to mark a line:

- Click on the line number located to the right of the code editor of the line you want to mark.
- Place the cursor on the line you want to mark and press `[Alt + M]`.

Unmarking a line will remove the highlight on the line and the check mark next to its line number. There are two ways to unmark a line:

- Click on the line number located to the right of the code editor of the currently marked line.
- Place the cursor on the marked line you want to unmark and press `[Alt + M]`.

For specific behaviors of marking and unmarking, please refer to section 4.1 on page 3.

3.2 Clear and Unclear

Clear will clear all currently marked lines from the current file. There are two ways to clear all marked lines:

- Choose **DeMark | Clear** in the toolbar.
- Press `[Alt + D + D]`.

Unclear will restore all previously cleared line for a file. There are two ways to unclear all previously cleared lines:

- Choose **DeMark | Unclear**.
- Press `[Alt + Shift + D + D]`.

For specific behaviors of clear and unclear, please refer to section 4.2 on page 3.

3.3 Toggle

The Toggle feature allows the user to comment/uncomment all the marked lines at a time.

- Choose **DeMark | Toggle**.
- Click the green and white toggle button located in the quick tools bar.

For specific behaviors of toggle, please refer to section 4.3 on page 4.

3.4 Display

Display allows the user to view all marked lines in a file from a more project overview level by only displaying the marked lines along with a range of lines above and below to provide context:

- Choose **DeMark | Display** in the toolbar.

For specific behaviors of display, please refer to section 4.4 on page 4.

4 Specific Behaviors of Functionality

4.1 Marking and unmarking:

- Marked lines will have a description that says “DeMark.”
- Marking and unmarking can only be done in IntelliJ and the marked lines will not show up as marked in other text editors or IDEs.
- Lines that are marked using DeMark will still be marked and visible in IntelliJ builds that do not have DeMark installed.
- Lines that are marked using DeMark will still be visible and editable in other text editors or IDEs. This includes insertion and deletion.
- Lines written in other files and marked using other IDEs will not show up as marked in IntelliJ but the line will still be there.
- Lines deleted in other IDEs or text editor will have the correct behavior as deleted in IntelliJ as well. This will also remove the mark for said line.

4.2 Clearing and unclearing:

- Clearing will remove all marked lines with descriptions containing “DeMark” from the current file, regardless of whether the file is commented or not.
- Clearing will not clear lines that do not have “DeMark” as a description. This means that bookmarks made for other purposes will not be removed.
- Clearing when there are no marked lines will produce no effect.
- Unclearing will restore all lines that were removed by the most recent clear function to the current file.
- Unclearing will not restore lines that were not removed by the clear function. This includes lines that were manually deleted by the user in IntelliJ, another text editor, or IDE; and lines that were deleted by other plugins or extensions.
- Unclearing multiple clears in a row will restore the most recent clear, the second most recent clear, and so on.
- Lines that are added and marked after a clear action can still be cleared. This and the previous unclear behavior allows for the following pattern of use and any similar: “**mark-clear-mark-clear-unclear-unclear.**”
- Lines that are added and marked after clear will not be removed by unclear. This allows for the following pattern of use and any similar: “**mark-clear-mark-unclear.**”

4.3 Toggle

Because toggling will put the plugin into two separate states, each state will have their own specific behaviors with some overlapping features.

4.3.1 Both states of toggle:

- Lines would still be able to be individually uncommented.
- Lines would still be able to be manually deleted.
- Lines deleted in other text editors or IDEs will still be represented correctly as deleted.

4.3.2 “Off” state:

- Marked lines that were already commented will gain an extra level of comment. This is similar to saying that the line will be commented twice.
- Marked lines that were commented will still be editable.
- Clearing and unclearing during this state is not permitted.
- Marking and unmarking in this state is permitted, the lines with their marking changed will retain their current levels of comment.

4.3.3 “On” state:

- Marked lines having more than one level of comment will lose one level.
- Marked lines that were uncommented during “off” state will stay uncommented.
- Clearing and unclearing in this state is permitted.
- Marking and unmarking in this state is permitted, the lines with their marking changed will retain their current levels of comment.

4.4 Display:

- Only marked lines for the current file will be displayed.
- Display will provide 3 extra lines of code above and below each marked line for context purposes.
- Displayed lines will be read-only. This includes the inability to mark, unmark, insert, or delete.

5 Reporting Bugs

If you find any bugs in our plugin, please submit an issue to our GitHub page: <https://github.com/DeFacto-UW/DeMark/issues> with the following information:

- IntelliJ Version located by going to **Help | About**
- A brief description of the bug.
- Specific steps to recreate the bug.
- *(Optional)* A screen shot of the bug.

References

- [1] *Installing, updating and uninstalling repository plugins*. Available at <https://www.jetbrains.com/help/idea/installing-updating-and-uninstalling-repository-plugins.html>.