

February, 2025

Sławomir Rogóż

Project Manager AI/ ML / R&D

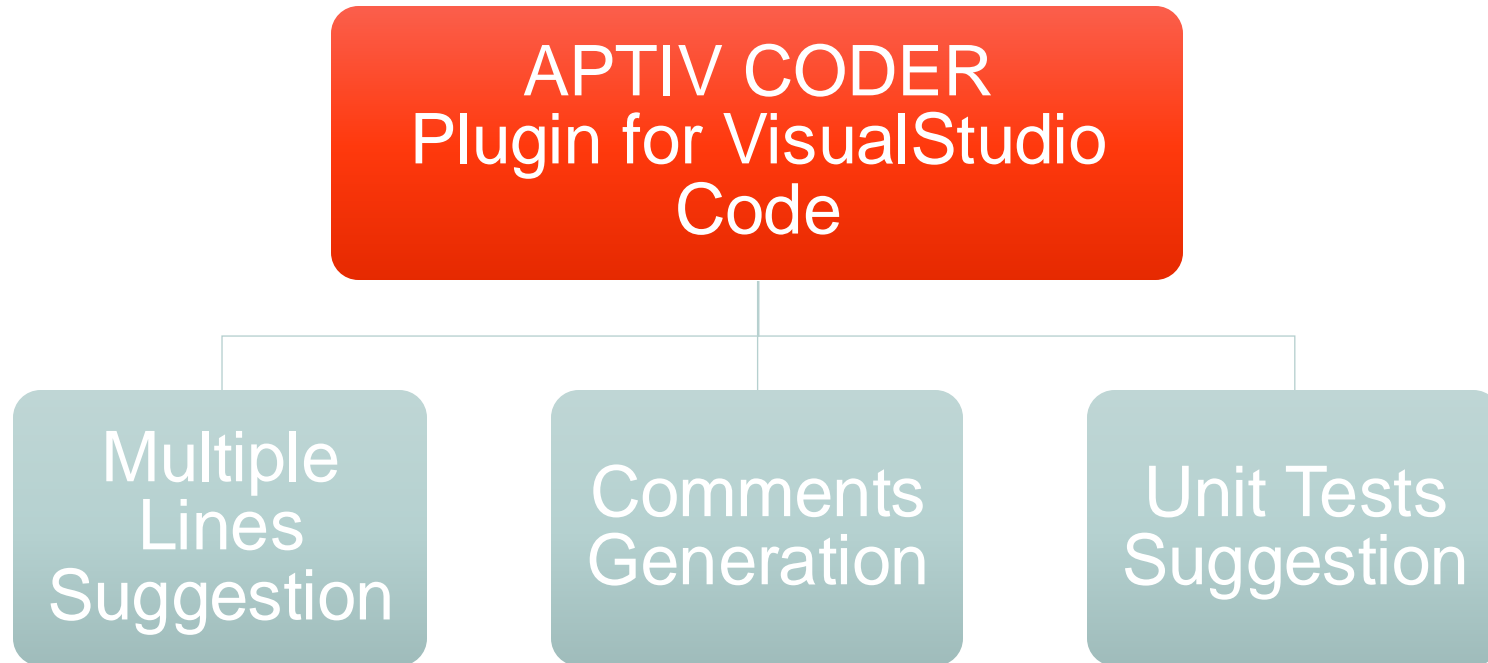


GEN AI APTIV CODER

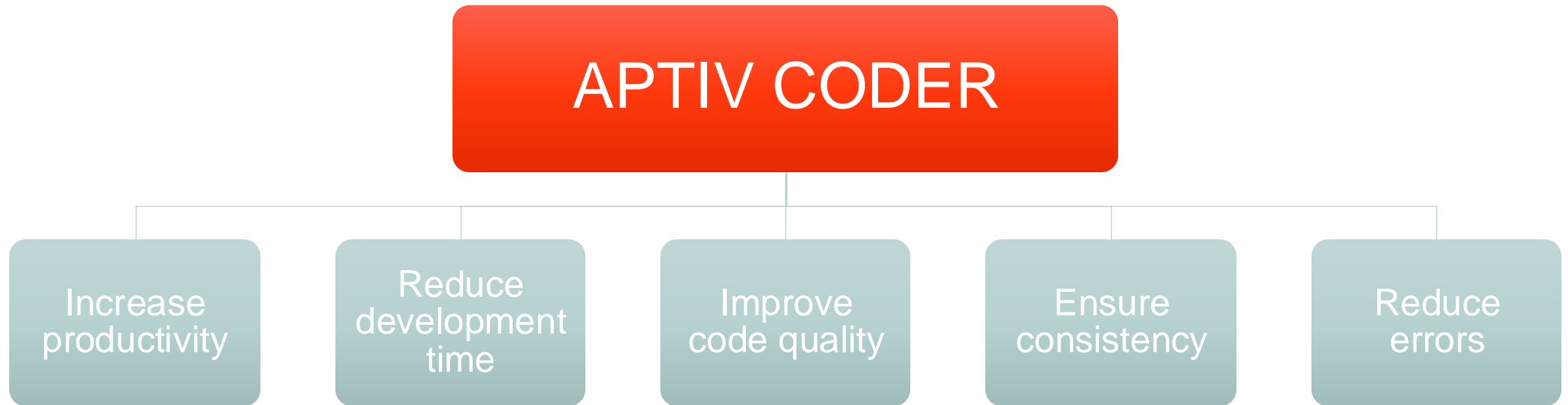
• APTIV •

APTIV CODER – Introduction

APTIV CODER - Introduction - What



APTIV CODER - Introduction - Why



APTIV CODER - What's in it for me

From perspective of code developers:

- **Increased Productivity:** AI code generation can automate repetitive and mundane tasks (both coding and unit tests generation), freeing up developers to focus on more complex and high-value tasks, leading to increased productivity and efficiency.
- **Reduced Development Time:** AI-generated code can reduce the time spent on coding, allowing developers to deliver projects faster and meet tight deadlines.
- **Improved Code Quality:** AI code generation can produce high-quality, optimized code that is less prone to errors and bugs, reducing the need for debugging and testing.
- **Consistency and Standardization:** AI-generated code can ensure consistency in coding styles and conventions, making it easier to maintain and update codebases.
- **Error Reduction:** AI-generated code can reduce errors and bugs, leading to higher-quality software and fewer issues in production.

APTIV CODER – Installation

APTIV CODER - How to access the tool?

To get started with Aptiv Coder, download the plugin from [Aptiv_Coder_Release](#) location and follow the installation steps.

If you have no access to the tool, submit „Domain Security Group - Add/Remove User” request in Service Now:

[Domain Security Group - Add/Remove User - Service Portal \(service-now.com\)](#) and fill out:

- *What would you like to do?* – **Add User To Domain Security Group**
- *Domain Security Group Name* – **SSO_APTIV_CODE_GEN_PROD**
- *Group Owner's Name* – Sławomir Rogóż (Krishna Koravadi)
- *Business Reason* – As per your demand justification

Access will be granted after ServiceNow ticket is approved and Security Team Task is closed.

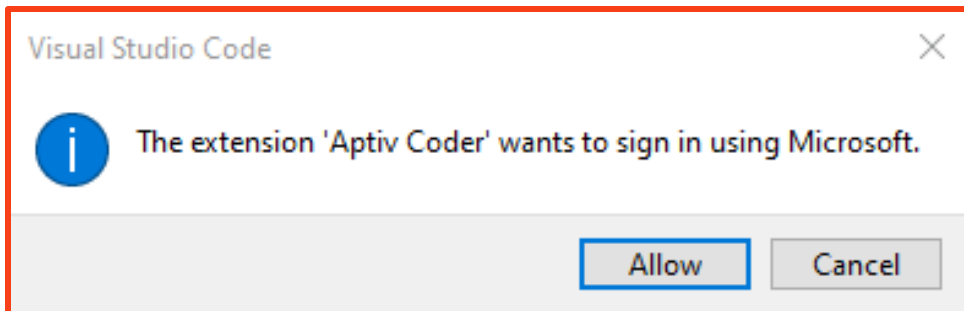
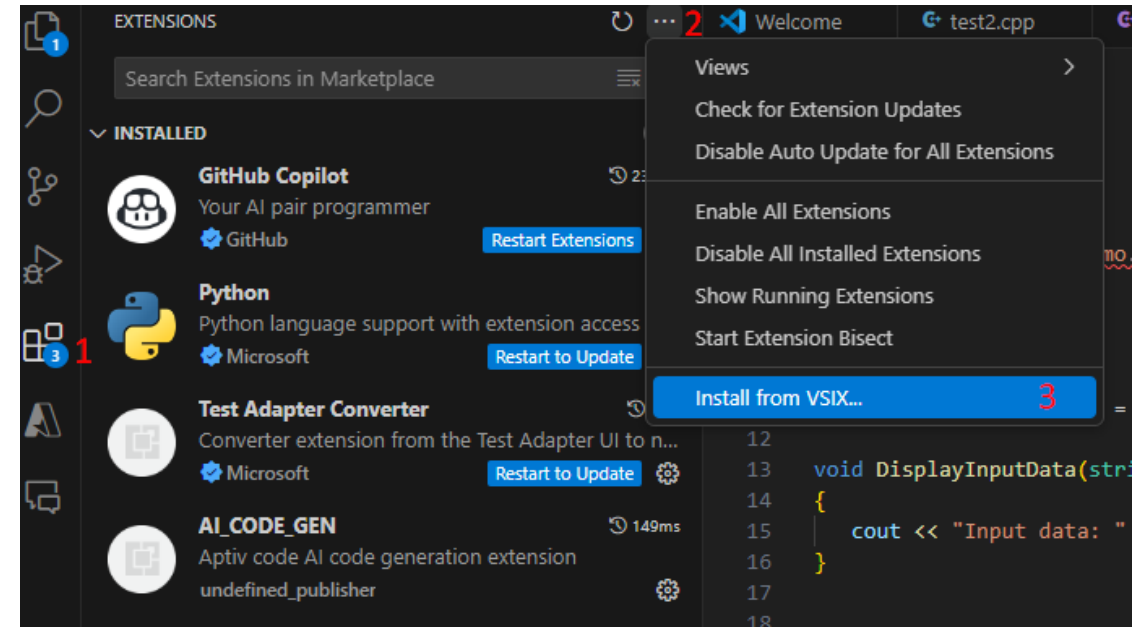
NOTE: *Please double check you are using VPN connection if you are not directly connected to internal company network while performing access process*

APTIV CODER - Installation

Installing Aptiv Coder plugin in VSCode:

1. Open the Extensions tab in VSCode.
2. Click the three dots and select „Install from VSIX...”.
3. Choose the *aptivcoder1.0.0.vsix* file and click the „Install” button.

The plugin is installed.



Authentication:

After first request Aptiv Coder will prompt you to sign in using Microsoft Authentication. Click the **Allow** button.

If you are part of the SSO production group, you can start using the tool.

APTIV CODER – Available features

After successful installation, press **Ctrl + Shift + P**, to access below commands and enhance your coding experience:

- Aptiv Coder: Complete multiple lines
- Aptiv Coder: Generate Comment
- Aptiv Coder: Generate Unit Test

APTIV CODER – Code suggestion

APTIV CODER - Code suggestion – input comment

Code generated based on function description.

To use this feature:

1. Provide a description of the function.
2. Place the cursor on a new line.
3. Select „*Complete multiple lines*” to generate the code.

INPUT:

```
34  /*Function to remove repeated values from vector and sort
35  Param1 - vector
36  */
37
38
39
40
41
42
```

OUTPUT:

```
34  /*Function to remove repeated values from vector and sort
35  Param1 - vector
36  */
37  void RemoveRepeatedValues(vector<int>& vec)
38  {
39      sort(vec.begin(), vec.end());
40      vec.erase(unique(vec.begin(), vec.end()), vec.end());
41  }
42
```

APTIV CODER - Code suggestion – function name

Code generated based on the function name only.

To use this feature:

1. Provide the name of the function.
2. Place the cursor at the end of the function name.
3. Select „*Complete multiple lines*” to generate the code.
4. The models uses the function name and context in the working file to generate the function implementation.

INPUT:

```
58 void RemoveRepeatedValuesDescendingSort
59
60
61
62
63
```

OUTPUT:

```
57
58 void RemoveRepeatedValuesDescendingSort(vector<int>& vec)
59 {
60     sort(vec.begin(), vec.end(), greater<int>());
61     vec.erase(unique(vec.begin(), vec.end()), vec.end());
62 }
63
```

APTIV CODER - Code suggestion – incomplete code

Code generated based on the incomplete code

To use this feature:

1. Place the cursor at the end of incomplete code.
2. Select „*Complete multiple lines*” to suggest rest of the implementation.
3. The model uses the code above the cursor as a context to suggest code.

```
int calculator(string input, char operation, int num1, int num2)
{
    int result = 0;
    switch (operation)
    {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
```

INPUT

OUTPUT

```
int calculator(string input, char operation, int num1, int num2)
{
    int result = 0;
    switch (operation)
    {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
        case '/':
            if (num2 != 0)
                result = num1 / num2;
            else
                cout << "Error: Division by zero!" << endl;
            break;
        default:
            cout << "Error: Invalid operation!" << endl;
            break;
    }
    return result;
}
```

APTIV CODER – Function description suggestion

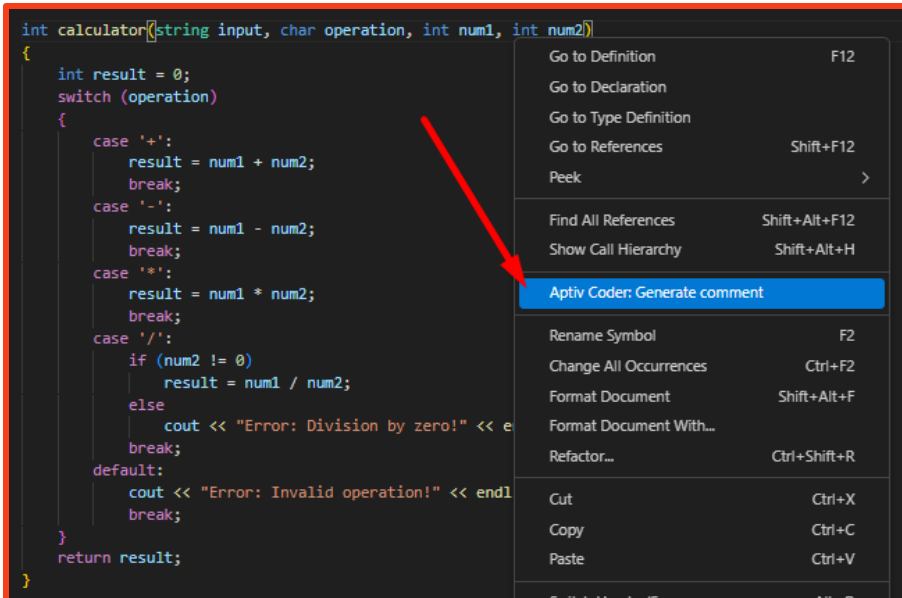
APTIV CODER - Function description suggestion

To use this feature:

1. Right-click on the code function and select: *Aptiv Coder: Generate comment*
2. Just above your function, the tool generates a description of it.

INPUT:

OUTPUT:



```
int calculator(string input, char operation, int num1, int num2)
{
    int result = 0;
    switch (operation)
    {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
        case '/':
            if (num2 != 0)
                result = num1 / num2;
            else
                cout << "Error: Division by zero!" << endl;
            break;
        default:
            cout << "Error: Invalid operation!" << endl;
            break;
    }
    return result;
}
```

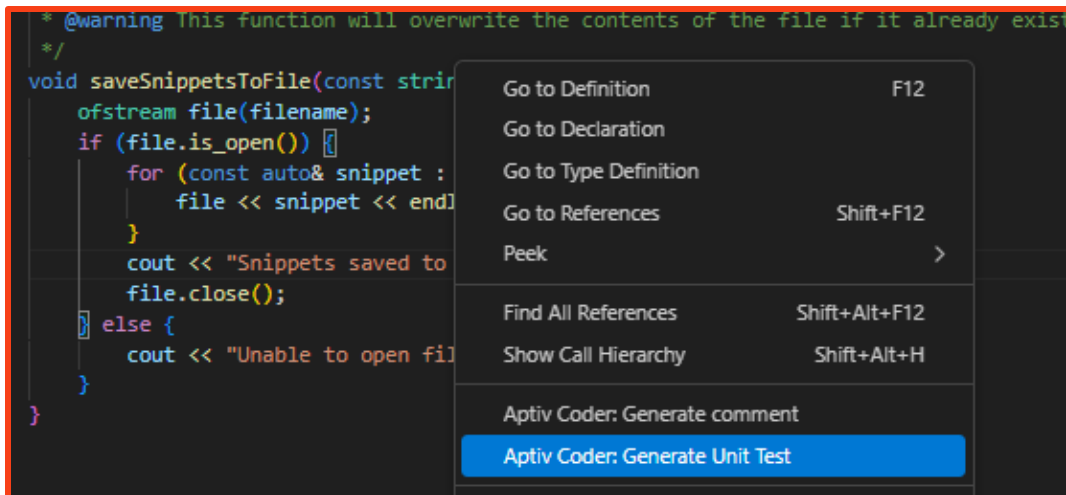
```
/**
 * Calculates the result of a mathematical operation between two numbers.
 *
 * @param input A string representing the input expression (not used in this implementation).
 * @param operation A character representing the mathematical operation to perform (+, -, *, /).
 * @param num1 The first number in the operation.
 * @param num2 The second number in the operation.
 *
 * @return The result of the mathematical operation.
 *
 * @note The input string is currently not used in this implementation.
 *       This function does not handle cases where num2 is zero for division operation.
 *       In case of invalid operation, it prints an error message and returns 0.
 *
 * Local variables:
 * - result: An integer to store the result of the mathematical operation.
 */
int calculator(string input, char operation, int num1, int num2)
```

APTIV CODER – Unit Tests Suggestion

APTIV CODER - Function description suggestion

To use this feature:

1. Right-click on the code function and select *Aptiv Coder: Generate Unit Test*
2. After a few seconds unit test cases to cover particular function are generated below the function implementation.



```
* @warning This function will overwrite the contents of the file if it already exist
*/
void saveSnippetsToFile(const string& filename) {
    ofstream file(filename);
    if (file.is_open()) {
        for (const auto& snippet : snippets) {
            file << snippet << endl;
        }
        cout << "Snippets saved to " << filename << endl;
        file.close();
    } else {
        cout << "Unable to open file " << filename << endl;
    }
}
```

Go to Definition F12
Go to Declaration
Go to Type Definition
Go to References Shift+F12
Peek >
Find All References Shift+Alt+F12
Show Call Hierarchy Shift+Alt+H
Aptiv Coder: Generate comment
Aptiv Coder: Generate Unit Test

INPUT:

OUTPUT:



```
}

#include <gtest/gtest.h>
#include <fstream>
#include <string>
#include <vector>

class SaveSnippetsTest : public ::testing::Test {
protected:
    void SetUp() override {
        codeSnippets = {"snippet1", "snippet2", "snippet3"};
    }

    void TearDown() override {
        remove("test_snippets.txt");
    }

    vector<string> codeSnippets;
};

TEST_F(SaveSnippetsTest, SavesSnippetsSuccessfully) {
    const string filename = "test_snippets.txt";
    saveSnippetsToFile(filename);

    ifstream file(filename);
    ASSERT_TRUE(file.is_open());

    string line;
    vector<string> savedSnippets;
    while (getline(file, line)) {
        savedSnippets.push_back(line);
    }
    file.close();

    EXPECT_EQ(codeSnippets, savedSnippets);
}
```

APTIV CODER – Code Markers

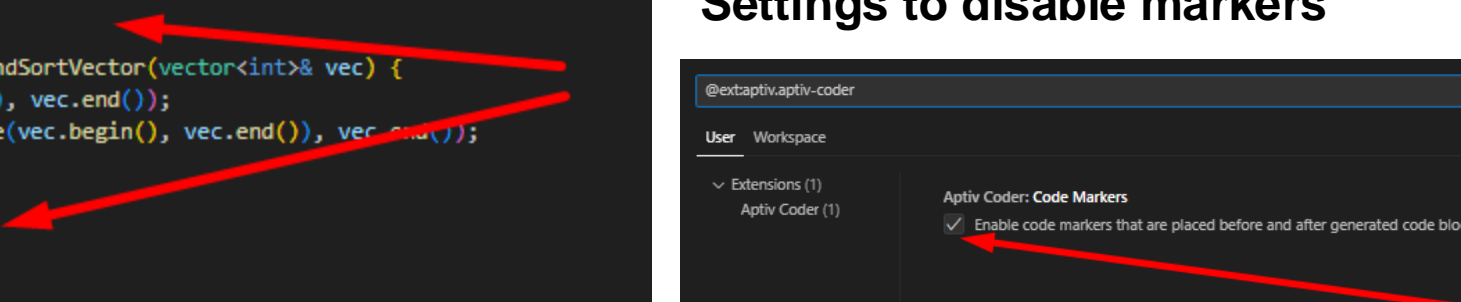
APTIV CODER - Code Markes

According to [Terms & Conditions](#), in default Aptiv Coder suggests code with special markers before and after an AI assisted block exceeding 200 of characters or 2 lines of code. Disabling this functionality is forbidden when working on production software. Snippet scanning must be carried out for each production software for all programs leveraging AI Code Assistance.

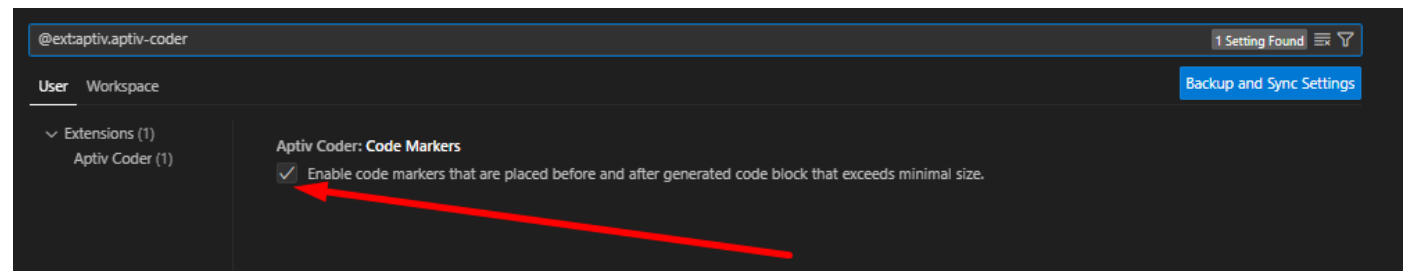
When working on non-production projects, the user can disable markers generation in the plugin settings.

Example of markers

```
/*Function to remove repeated values from vector and sort
Param1 - vector
*/
//START_APTIV_CODER
void RemoveRepeatsAndSortVector(vector<int>& vec) {
    sort(vec.begin(), vec.end());
    vec.erase(unique(vec.begin(), vec.end()), vec.end());
}
//END_APTIV_CODER
```



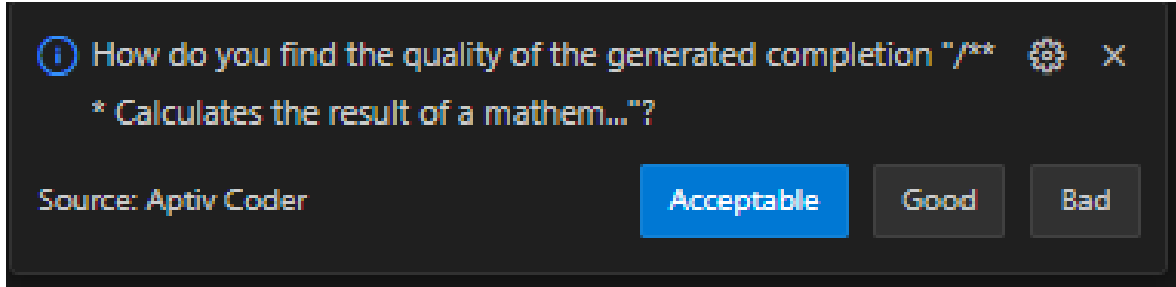
Settings to disable markers



APTIV CODER – User Feedback and Releases Notification

APTIV CODER - User Feedback

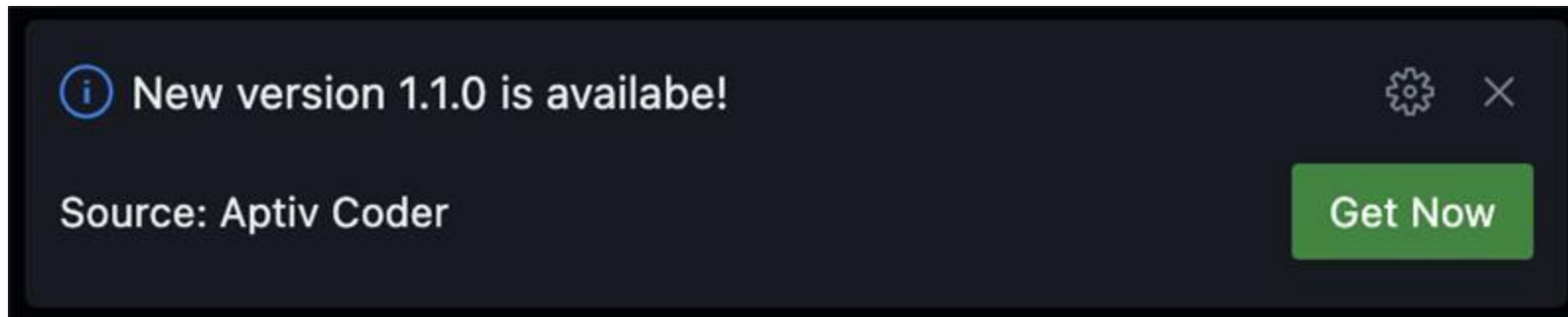
After each request feedback window pops out in the IDE's bottom corner.



Please provide feedback as frequently as possible. Your input will help us improve the tool for the future.

APTIV CODER – Stay Up-to-Date

Starting from release 1.1.0, you will be notified about the newest versions of Aptiv Coder, ensuring you always have access to the latest features and improvements.



APTIV CODER – How to report an issue?

APTIV CODER - How to report issue?

Raise an Issue via SNOW (Service Now):

- Use the **Virtual Agent** directly in MS Teams:

<https://teams.microsoft.com/l/app/d6502493-6f10-4594-a5c5-5c7e304acc66?source=app-details-dialog>

- Use a **Generic Form** in Service Now ("Get help") to raise an issue:
[Home Page - Service Portal \(service-now.com\)](https://service-now.com/home)

- Select AptivCoder application:

If your issue pertains to an specific application from the below list kindly choose that particular applications from the options. ?

If your issue is not related to the applications on this list, please select –None—from the list. x

GEN AI APTIVCODER | ADM0004200



Global IT Service Desk

ServiceNow

Open



Get Help

Report your IT related issues.