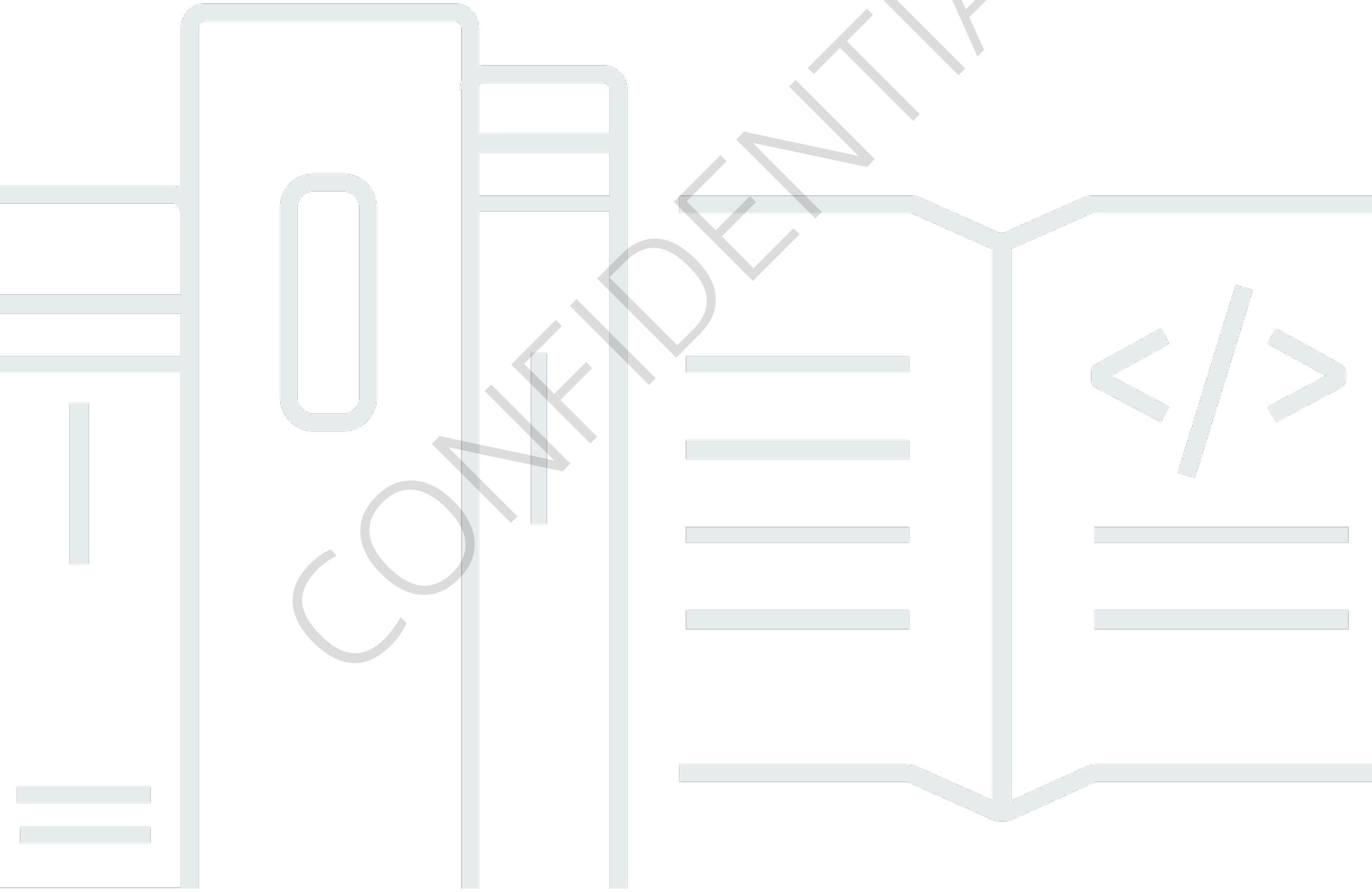
User Guide

Amazon Q Developer



Amazon Q Developer: User Guide

This prerelease documentation is confidential and is provided under the terms of your nondisclosure agreement with Amazon Web Services (AWS) or other agreement governing your receipt of AWS confidential information.

Transforming .NET applications

With the Amazon Q Developer agent for code transformation, you can port your Windows-based .NET applications to cross-platform .NET applications through a generative-AI powered refactoring workflow. It also helps you upgrade outdated versions of cross-platform .NET applications to supported versions.

To transform a .NET solution or project, Amazon Q analyzes your codebase, determines the necessary updates to port your application, and generates a transformation plan before the transformation begins. During this analysis, Amazon Q divides your .NET solution or project into decomposable buildable units (DBUs) that you can view in the transformation plan. A DBU is a project and all its dependencies taken together to generate a buildable unit of code such as a dynamic link library (DLL) or an executable.

During the transformation, Amazon Q provides step-by-step updates in a hub where you can monitor progress. After transforming your application, Amazon Q generates a summary with the proposed changes in a diff view for you to verify the changes before accepting them. When you accept changes, Amazon Q can make in-place updates to your .NET solution or project.

Amazon Q performs four keys tasks to port .NET applications to Linux:

- Upgrades language version – Replaces outdated C# versions of code with Linux-compatible C# versions
- Migrates from .NET Framework to cross-platform .NET – Migrates projects and packages from Windows dependent .NET Framework to cross-platform .NET compatible with Linux
- Rewrites code for Linux compatibility – Refactors and rewrites deprecated and inefficient code components
- Generates a Linux compatibility readiness report – For open-ended tasks where user intervention is needed to make the code build and run on Linux, Amazon Q provides a detailed report of actions needed to configure your application after transformation

For more information about how Amazon Q performs .NET transformations, see How it works.

Topics

- Step 1: Prerequisites
- Step 2: Transform your application
- How the Amazon Q Developer agent for .NET code transformation works
- Troubleshooting code transformation

Step 1: Prerequisites

Before you continue, make sure you've completed the steps in Set up Amazon Q in your IDE.

Make sure that the following prerequisites for your application are met before you begin a .NET transformation job:

- Your application contains only .NET projects written in C#.
- Your application only has Microsoft-authored NuGet package dependencies
- If your application is dependent on Internet Information Services (IIS), only default IIS configurations are used
- Amazon Q will evaluate the type of the project you selected and its dependencies to create a DBU. A DBU can have the following project types:
  - Console application
  - Class library
  - Web API
  - WCF Service
  - Business logic layers of Model View Controller (MVC) and Single Page Application (SPA)

> **Note**
>
> Amazon Q doesn't support transforming View components. If Amazon Q detects Views in your selected solution or project, it will transform your selection partially by excluding View components, and you might need to refactor further to make your code buildable on the target .NET version.

Step 2: Transform your application

To transform your .NET solution or project, complete the following procedure:

1. Open any C# based solution or project in Visual Studio that you want to transform.
2. Choose Solution Explorer on the right side of the Visual Studio window.
3. From the Solution Explorer, right click a solution or project you want to transform, and then choose **Transform a solution or project with Amazon Q Developer**.

4. The **Transform a solution or project with Amazon Q Developer** window appears.

   The solution or project you selected will be chosen in the **Choose a solution or project to transform** dropdown menu. You can expand the menu to choose a different solution or project to transform.

   In the **Select a .NET target version** dropdown, choose the .NET version you want to upgrade to.
5. Choose **Confirm** to begin the transformation.
6. Amazon Q begins transforming your code. You can view the Transformation Plan it generates for details about how it will transform your application.

   A Transformation Hub opens where you can monitor progress for the duration of the transformation. After Amazon Q has completed step 2, **Awaiting job transformation startup**, you can navigate away from the project for the duration of the transformation.
7. After the transformation is complete, navigate to the Transformation Hub and choose **View diffs** to review the proposed changes from Amazon Q in a diff view.
8. Choose **View code transformation summary** for details about the changes Amazon Q made. You can also download the Transformation summary by choosing **Download summary as .md**.

   If any of the items in the **Decomposable buildable units** table require input under the Linux porting status, you must manually update some files to run your application on Linux.

   a. From the Actions dropdown menu, choose Download Linux readiness report.
   b. A .csv file opens with any changes to your project that you must complete before your application is Linux compatible. It includes the project and file that need to be updated, a description of the item to be updated, and an explanation of the issue. Use the **Recommendation** column for ideas on how to address a Linux readiness issue.
9. To update your files in place, choose Accept changes from the Actions dropdown.

How the Amazon Q Developer agent for .NET code transformation works

Review the following sections for details about how .NET transformation works.

Analyzing your application and generating a transformation plan

Before a transformation begins, Amazon Q locally builds your code to ensure it’s buildable and configured correctly for transformation. Amazon Q then uploads your code to a secure build environment on AWS, analyzes your codebase, and determines the necessary updates to port your application.

During this analysis, Amazon Q divides your .NET solution or project into decomposable buildable units (DBUs). A DBU is a project and all its dependencies that together generate a buildable unit of code such as a dynamic link library (DLL) or an executable. Even if you didn't select all project dependencies to be transformed, Amazon Q determines the dependencies needed to build your selected projects and transforms them too, so that your transformed application will be buildable and ready for use.

After analyzing your code, Amazon Q generates a transformation plan that outlines the proposed changes that Amazon Q will make, including a list of DBUs and their dependencies that will be transformed.

Transforming your application

To start the transformation, Amazon Q builds your code again in the secure build environment to ensure it's buildable remotely. Amazon Q then begins porting your application. It works from the bottom up, starting with the lowest level dependency. If Amazon Q runs into an issue with porting a dependency, it stops the transformation and provides information about what caused the error.

The transformation includes the following updates to your application:

- Replacing outdated C# versions of code with Linux-compatible C# versions
- Upgrading .NET Framework to cross-platform .NET, including:
  - Identifying and iteratively replacing packages, libraries, and APIs
  - Upgrading and replacing NuGet packages and APIs
  - Transitioning to cross-platform runtime
  - Setting up middleware and updating runtime configurations
  - Replacing private or third party packages
  - Handling IIS and WCF components
  - Debugging build errors
- Rewriting code for Linux compatibility, including refactoring and rewriting deprecated and inefficient code to port existing code

Reviewing transformation summary and accepting changes

After the transformation is complete, Amazon Q provides a transformation summary with information about the proposed updates it made to your application, including the number of files changed, packages updated, and APIs changed. It flags any unsuccessful transformations, including affected files or portions of files and the errors encountered during an attempted build. You can also view a build summary with build logs to learn more about what changes were made.

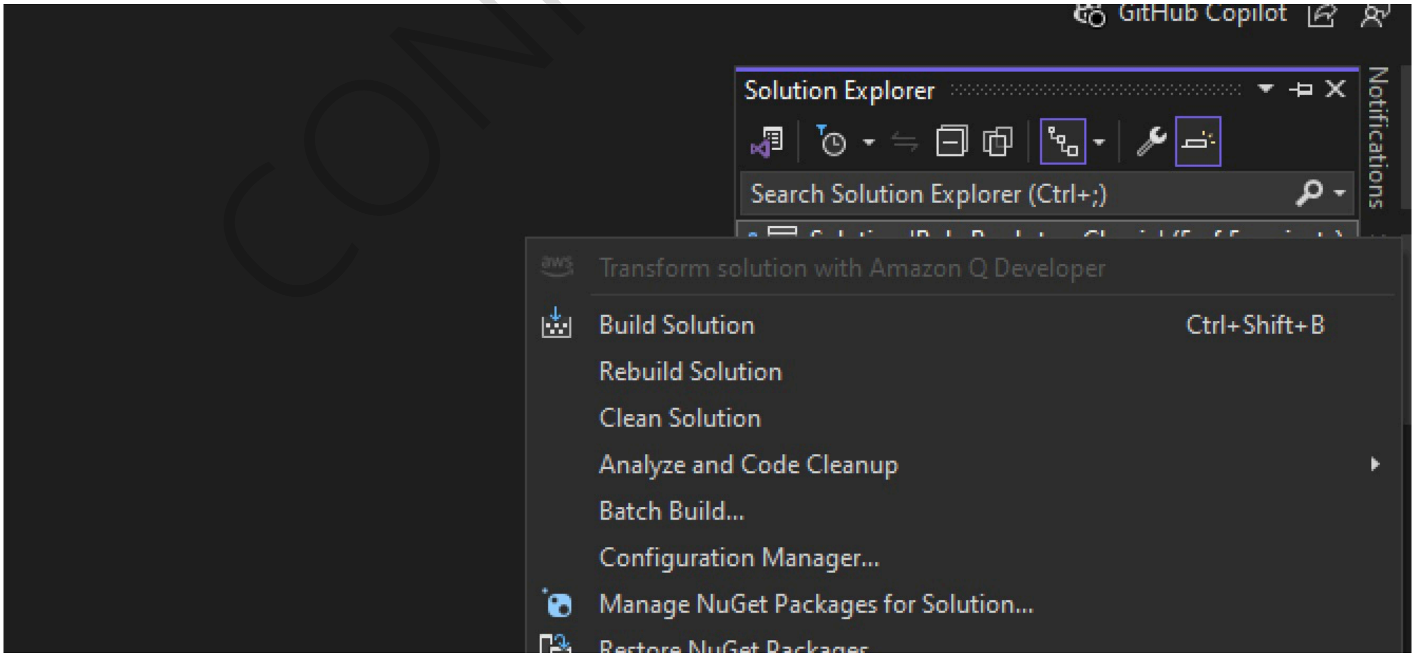
The transformation summary also provides a Linux porting status, which describes whether or not additional user input is needed to make the application Linux compatible. If any of the items in a DBU require input from you, you download a Linux readiness report that contains Windows-specific considerations that Amazon Q could not address at build time. These changes must be made manually before your application can be run on Linux.

You can review the proposed changes Amazon Q made in a diff view before accepting them as in-place updates to your files. After updating your files and addressing any items in the Linux readiness report, your application is ready to run on cross-platform .NET.

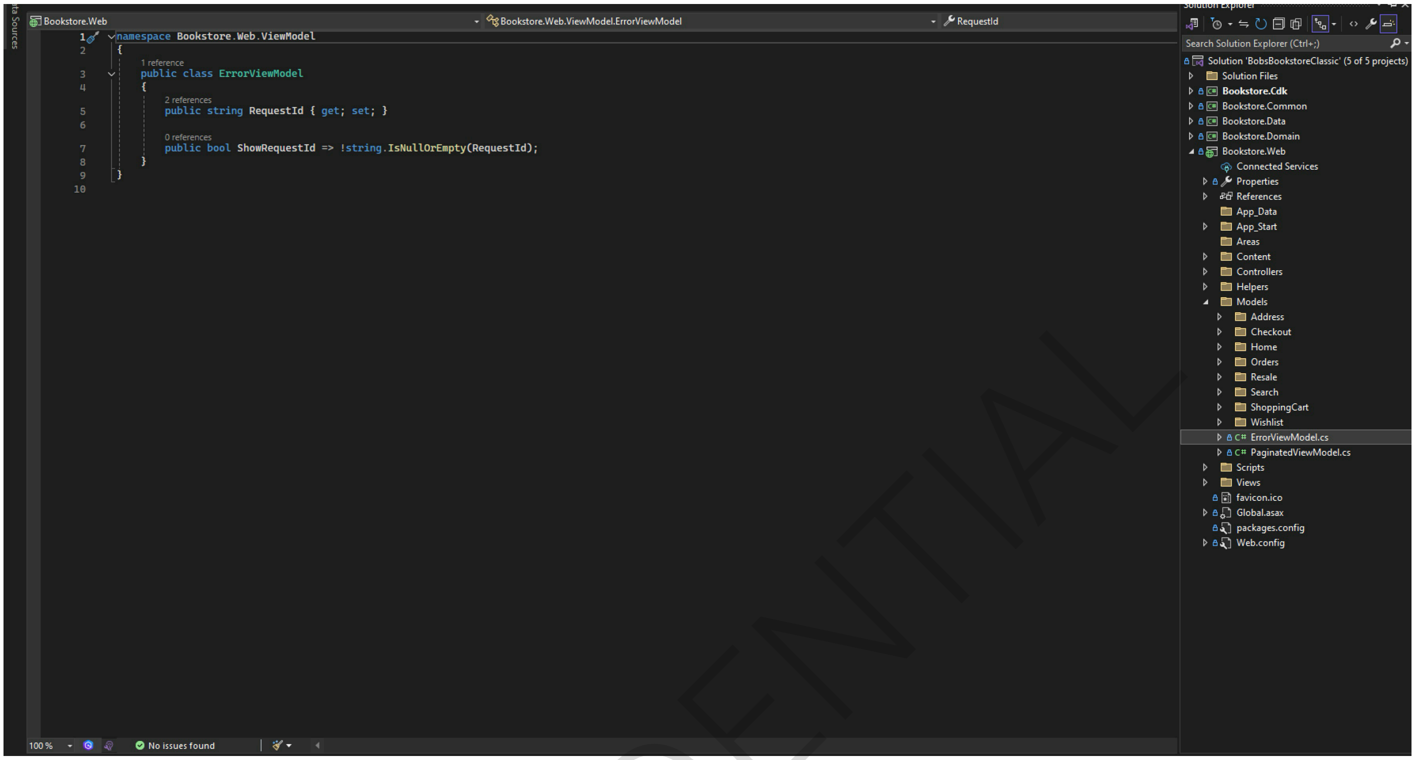
Troubleshooting code transformation

Use the following section to troubleshoot common issues with Amazon Q Developer.

- **Issue: The transform solution with Amazon Q Developer is grayed out as shown in the example below.**



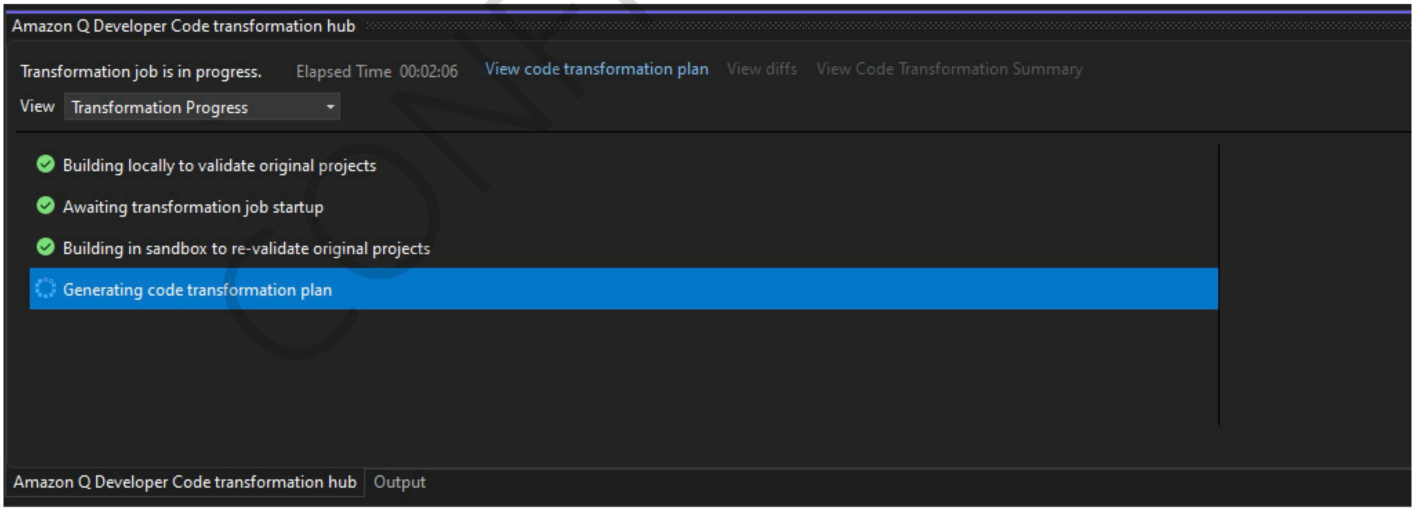
Solution: The Q Developer end-point is not yet active. Open any code file (`*.cs file`) and place your curser on it to activate Q Developer. The red exclamation mark next to Q logo will disappear once the Q Developer endpoint is connected.



```
namespace Bookstore.Web.ViewModel
{
    1 reference
    public class ErrorViewModel
    {
        2 references
        public string RequestId { get; set; }

        0 references
        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
    }
}
```

- **Issue: The transformation hub is spending a lot of time on a specific step. How do I know if a job is still progressing?**



Amazon Q Developer Code transformation hub

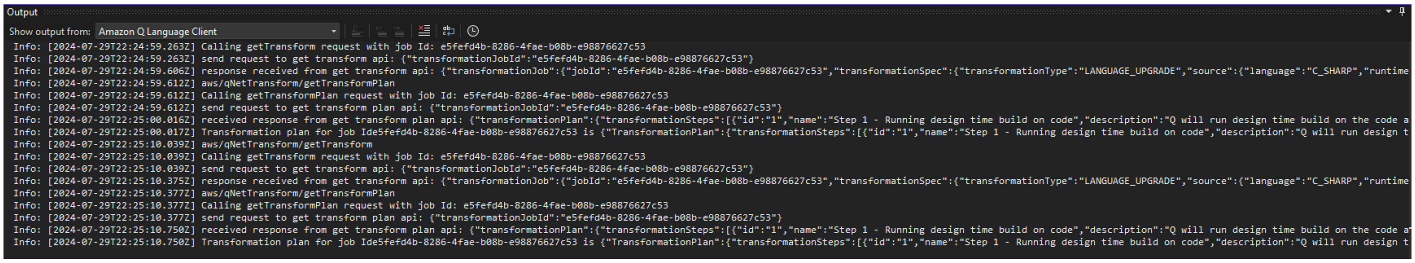
Transformation job is in progress. Elapsed Time 00:02:06

View: Transformation Progress

- ✓ Building locally to validate original projects
- ✓ Awaiting transformation job startup
- ✓ Building in sandbox to re-validate original projects
- Generating code transformation plan

View code transformation plan
View diffs
View Code Transformation Summary
Amazon Q Developer Code transformation hub
Output

To verify if the job is still active, click on the Output tab and select Amazon Q Language Client in the "Show output from:" dropdown menu. If the job is progressing, you will see diagnostic messages being added here.



```
Info: [2024-07-29T22:24:59.263Z] Calling getTransform request with job Id: e5fefd4b-8286-4fae-b08b-e98876627c53
Info: [2024-07-29T22:24:59.263Z] send request to get transform api: {"transformationJobId":"e5fefd4b-8286-4fae-b08b-e98876627c53"}
Info: [2024-07-29T22:24:59.606Z] response received from get transform api: {"transformationJob":{"jobId":"e5fefd4b-8286-4fae-b08b-e98876627c53","transformationSpec":{"transformationType":"LANGUAGE_UPGRADE","source":{"language":"C_SHARP","runtime
Info: [2024-07-29T22:24:59.612Z] aws/qNetTransform/getTransformPlan
Info: [2024-07-29T22:24:59.612Z] Calling getTransformPlan request with job Id: e5fefd4b-8286-4fae-b08b-e98876627c53
Info: [2024-07-29T22:24:59.612Z] send request to get transform plan api: {"transformationJobId":"e5fefd4b-8286-4fae-b08b-e98876627c53"}
Info: [2024-07-29T22:25:00.016Z] received response from get transform plan api: {"transformationPlan":{"transformationSteps":[{"id":"1","name":"Step 1 - Running design time build on code","description":"Q will run design time build on the code a
Info: [2024-07-29T22:25:00.017Z] Transformation plan for job Ide5fefd4b-8286-4fae-b08b-e98876627c53 is {"TransformationPlan":{"transformationSteps":[{"id":"1","name":"Step 1 - Running design time build on code","description":"Q will run design t
Info: [2024-07-29T22:25:10.039Z] aws/qNetTransform/getTransform
Info: [2024-07-29T22:25:10.039Z] Calling getTransform request with job Id: e5fefd4b-8286-4fae-b08b-e98876627c53
Info: [2024-07-29T22:25:10.039Z] send request to get transform api: {"transformationJobId":"e5fefd4b-8286-4fae-b08b-e98876627c53"}
Info: [2024-07-29T22:25:10.375Z] response received from get transform api: {"transformationJob":{"jobId":"e5fefd4b-8286-4fae-b08b-e98876627c53","transformationSpec":{"transformationType":"LANGUAGE_UPGRADE","source":{"language":"C_SHARP","runtime
Info: [2024-07-29T22:25:10.377Z] aws/qNetTransform/getTransformPlan
Info: [2024-07-29T22:25:10.377Z] Calling getTransformPlan request with job Id: e5fefd4b-8286-4fae-b08b-e98876627c53
Info: [2024-07-29T22:25:10.377Z] send request to get transform plan api: {"transformationJobId":"e5fefd4b-8286-4fae-b08b-e98876627c53"}
Info: [2024-07-29T22:25:10.750Z] received response from get transform plan api: {"transformationPlan":{"transformationSteps":[{"id":"1","name":"Step 1 - Running design time build on code","description":"Q will run design time build on the code a
Info: [2024-07-29T22:25:10.750Z] Transformation plan for job Ide5fefd4b-8286-4fae-b08b-e98876627c53 is {"TransformationPlan":{"transformationSteps":[{"id":"1","name":"Step 1 - Running design time build on code","description":"Q will run design t
```

Developing software with the Amazon Q Developer Agent for software development

The Amazon Q Developer Agent for software development can help you develop code features or make code changes to projects in your integrated development environment (IDE). You explain the feature you want to develop, and Amazon Q uses the context of your current project to generate an implementation plan and the accompanying code to implement the feature. Amazon Q can help you build AWS projects or your own applications.

You can start an entirely new project, or work on an open project in your IDE. When you develop in an existing project, Amazon Q uses all files in your workspace root as context to develop a plan and generate code.

To get started, open up a new or existing project and enter **`/dev`** in the Amazon Q chat panel. A new chat tab opens where you interact with Amazon Q to generate an implementation plan and new code for your feature.

Topics

Develop features with /dev

To develop a feature in your IDE, complete the following steps.

1. In your IDE, open a new or existing project or workspace where you want to develop features.