Application Note

C2000 IDE Assist Migration Tool Guide

Shashank Madineni, Nima Eskandari, Delaney Woodward, and Aishwarya Rajesh

ABSTRACT

C2000 IDE Assist, or C2000 IDEA, is an integrated development tool designed to enhance development for Texas Instruments' C2000™ microcontrollers. This provides a centralized environment within Visual Studio Code™ and Code Composer Studio™, offering features such as project detection, targeted collateral delivery, and developer efficiency tools. One of the key capabilities is migration support, which helps developers transition application codes between different C2000 devices efficiently.

This application note guides users through each step of the migration process, from detecting and setting up projects to executing migration checks and resolving compatibility issues. This document covers quick fixes for common migration concerns, migration reports for analyzing code differences, and important constraints and considerations to make sure of a smooth and reliable transition across device architectures.

Table of Contents

List of Figures

List of Tables

Trademarks

C2000™ and Code Composer Studio™ are trademarks of Texas Instruments.
Visual Studio Code™ is a trademark of Microsoft Corporation.
All trademarks are the property of their respective owners.

1 Introduction

The Texas Instruments' C2000 software development kits (SDK) provide a cohesive suite of software and documentation designed to accelerate development of real-time control applications. These resources include device-specific drivers, libraries, and peripheral examples, helping developers streamline projects. The SDKs are built to accommodate users of all experience levels, with beginner-friendly features to guide first-time users. The C2000 SysConfig tool, included in the SDKs, enhances the setup process by simplifying device and peripheral complexities through an intuitive visual interface. To further improve the initialization, run-time, and debug user experience, the C2000 IDEA centralizes tools and collateral within a single environment for a more efficient workflow.

This document describes how to get started with the C2000 IDE Assist Migration feature and effectively utilise to simplify and accelerate migration across C2000 devices.

1.1 Getting Started

This section introduces the basics of the C2000 IDEA tool, such as how to set up the development environment to utilize the extension.

- Download and install the latest offline installer for Code Composer Studio (CCS), version 20.0.0 or later.
- Launch CCS and navigate to the *Extensions* tab located in the left sidebar.
- Search for and install the C2000 IDEA Extension.
- Restart CCS to enable the extension. Users now see the C2000 IDEA icon in the left sidebar.
- Open a new or existing workspace as normal in CCS.

For detailed setup instructions and additional guidance, refer to the C2000 IDE Assist Tool Features Guide application note.

2 Overview

This section provides an overview of the C2000 IDEA interface and explains how to navigate the tool with ease. The C2000 IDEA - FEATURES panel, located on the left side of the screen, serves as an access for all functions of the tool. The following image illustrate this panel using a sample C2000WARE example to help users get familiar with the layout and options.

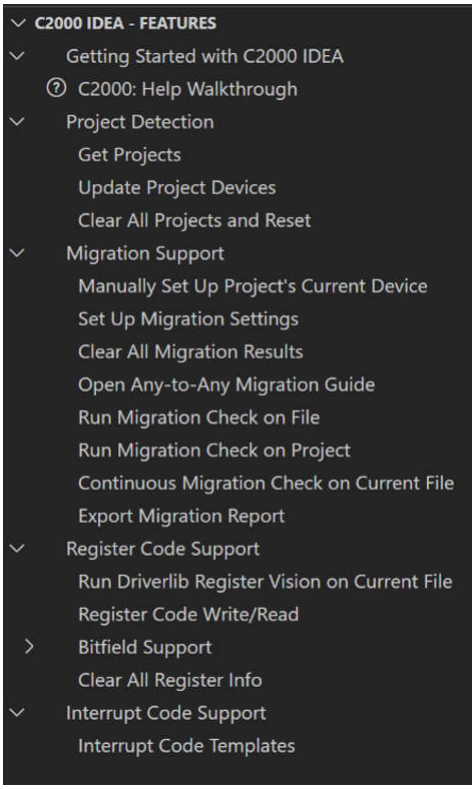

Figure 2-1. C2000 IDEA - FEATURES Panel

Within this panel, a dedicated *Migration Support* section offers quick and easy access to all migration-related features offered by the tool. Users can refer to the *Open Any-to-Any Migration Guide* HTML page as shown in Figure 2-2 and select the appropriate C2000ware version for both the current and target device. This enables the user to view driverlib code differences and explore one-to-one mappings across different C2000 devices.

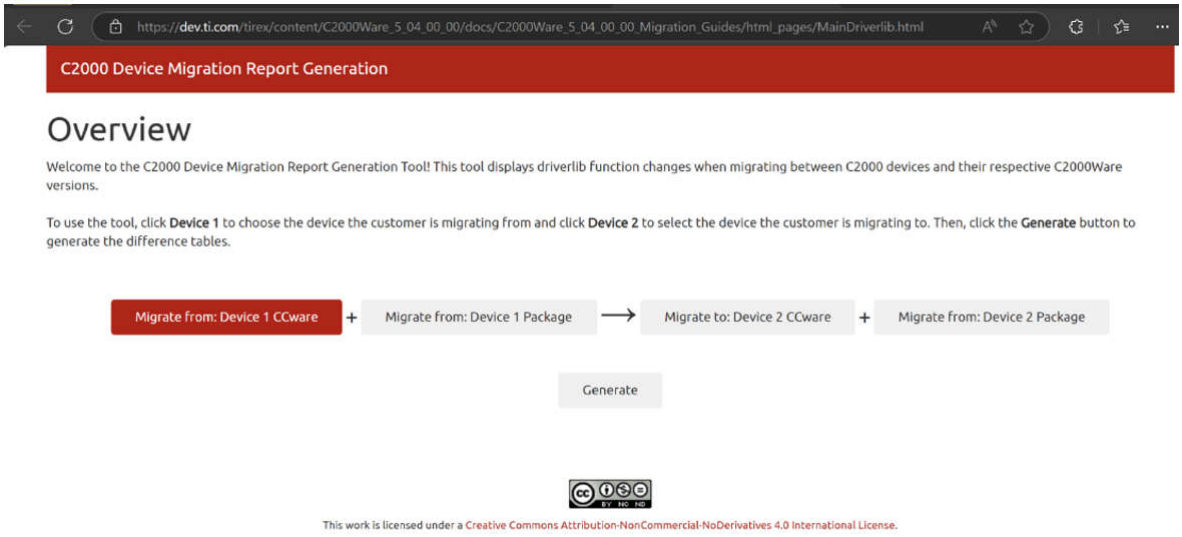

Figure 2-2. Any-to-Any Device Driverlib Migration HTML Page

3 Migration Support Feature

The C2000 IDEA tool simplifies code migration between C2000 devices by automatically identifying and highlighting changes in driverlib code, including added, removed, or modified Application Peripheral Interfaces (APIs), registers, fields, enums, and macros. This automation reduces manual effort, minimizes the risk of errors, and verifies a smooth and efficient migration process. The automation supports migration for both complete projects and standalone driverlib files, providing flexibility to meet various user requirements. For specific devices, the tool also enables bitfield code migration. C2000 IDEA simplifies migration between F28x-to-F28x and F28x-to-F29x devices by detecting architectural differences and offering targeted recommendations for a more reliable transition. Table 3-1 provides a clear summary of migration support across C2000 devices, helping users quickly assess compatibility and available support for both Driverlib and Bitfield implementations.

Table 3-1. Migration Support for C2000 Devices

Device Families	Driverlib Migration Support	Bitfield Migration Support
F29H85x	Yes	No
F28P55x	Yes	No
F28P65x	Yes	No
F28002x	Yes	No
F28004x	Yes	No
F28003x	Yes	No
F280013x	Yes	Yes (To)
F280015x	Yes	No
F2838x	Yes	No
F2837x	Yes	No
F2807x	Yes	No
F2803x	No	Yes(From)

This tool supports source files in specific file formats, as illustrated in Table 3-2.

Table 3-2. Migration Support for Various File Formats

Source File Format	Migration Support
*.c	Yes
*.h	Yes
*.asm	No
*.cmd	No
*.lib	No
Linker Files	No

Figure 3-1 outlines the key steps required to migrate a standalone file or an entire project, to make sure of a seamless and efficient migration process

Figure 3-1. C2000 IDEA - Migration Flow

3.1 Project Detection

All projects using C2000 devices can be detected by clicking the *Get Projects* element. This step allows the extension to recognize the device for each project and keeps track of which project each file belongs to. Project Detection unlocks the complete capabilities of the extension in real-time such as Driverlib Migration across devices. Some extension features are only available with project detection (or default device setup), whereas other features can be run on single files by requiring input of the current device.

Follow these steps to detect a project:

1. To detect all the projects in the workspace, use one of the two options below:
   a. Enter CTRL+SHIFT+P and click *C2000: Get Projects*.
   b. Click *Project Detection* > *Get Projects* in the *C2000 IDEA - Features* pane of the Extension tree.
2. Once run, all the projects in the workspace detected by the extension are present in the *C2000 IDEA - Projects* pane of the Extension tree. View the project and verify the device variant and current device details are as expected.

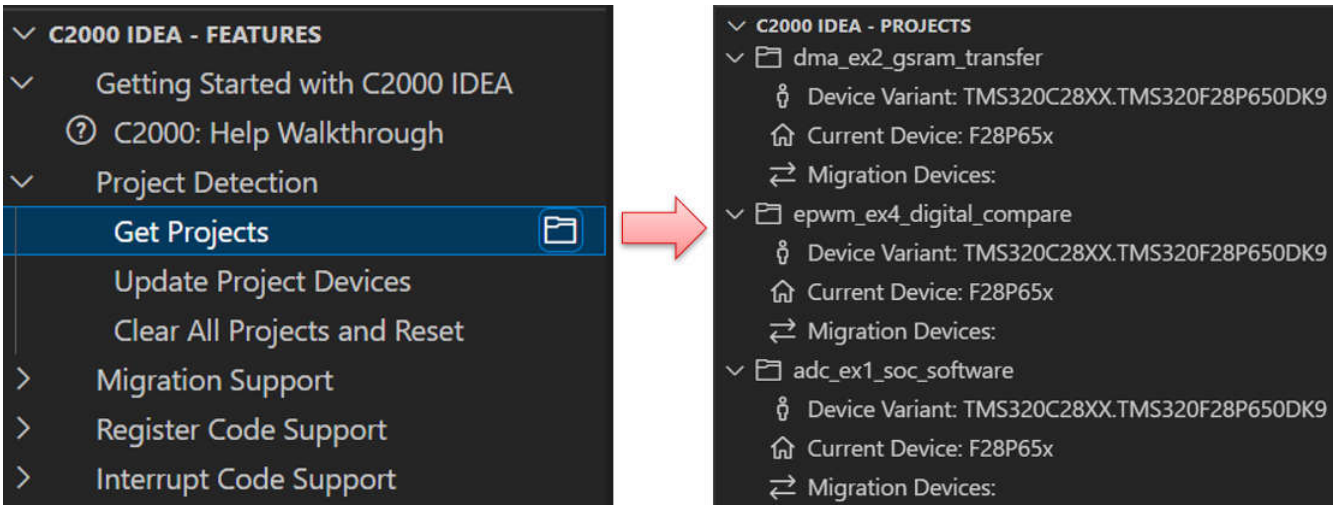

Figure 3-2. C2000 IDEA- Project Detection

3.2 Setup Migration Settings Page

This page provides a comprehensive interface for customizing the migration process across an entire project, allowing users to configure migration settings according to the specific requirements.

To access this page, users can use one of the following methods:

1. Enter CTRL+SHIFT+P and click *C2000: Set Up Migration Settings*.
2. Extension tree methods:
   a. Navigate to Migration Support > Set Up Migration Settings in the *C2000 IDEA - Features* pane.
   b. Click the icon next to Migration Devices under the detected project in the *C2000 IDEA - Projects* pane as shown in Figure 3-3.

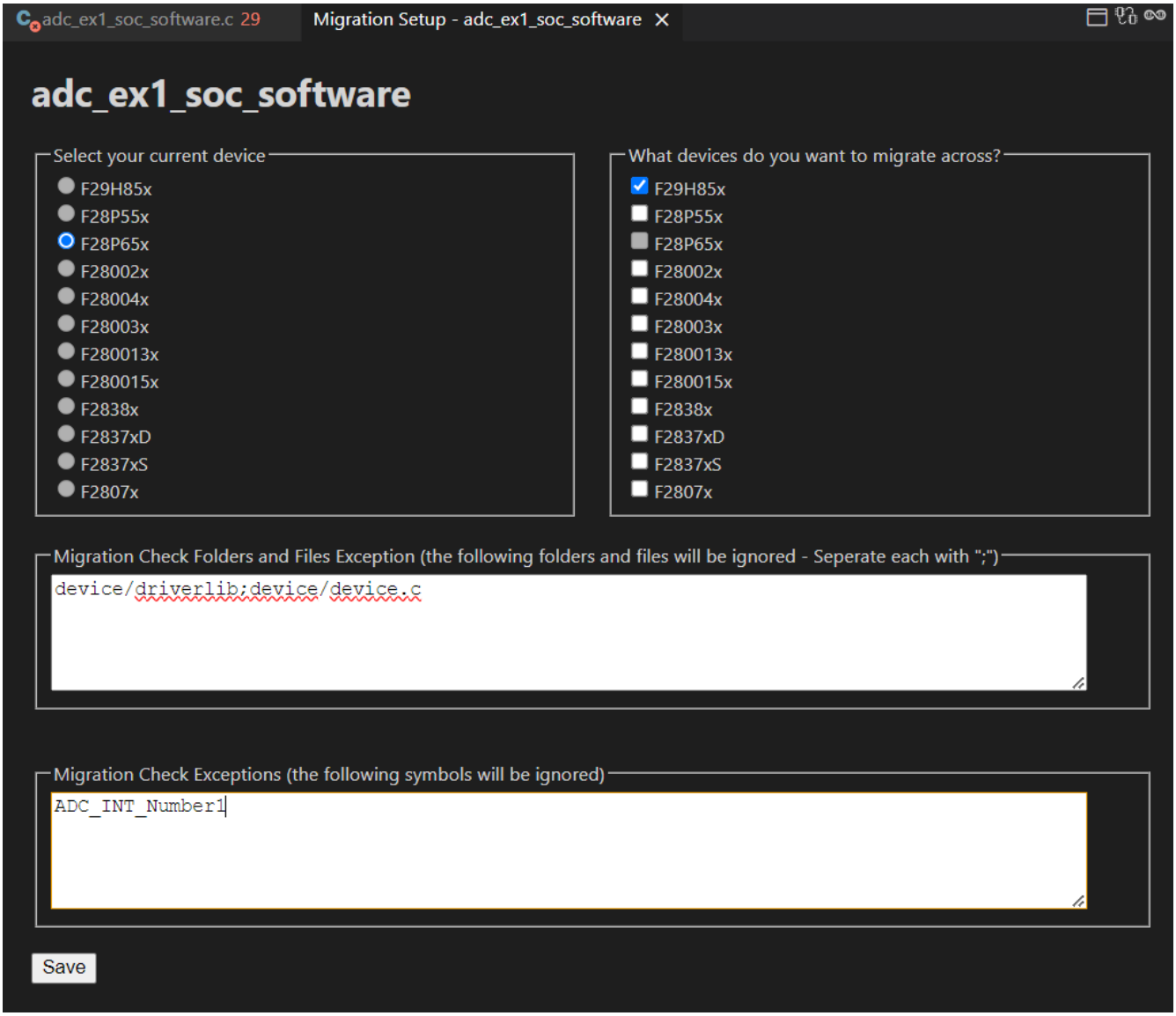

Figure 3-3. Migration Settings Setup Page

The Migration Settings page automatically detects and identifies the current device based on the project detection process completed in the previous step. Since the device is predetermined, users cannot modify directly within the Migration Settings page.

If a change to the current device of the project is required, then users can do so through the following methods:

1. Enter CTRL+SHIFT+P and click *C2000: Manually Set Up Project's Current Device*.
2. Click the icon next to *Current Device* under the detected project in the *C2000 IDEA - Projects* pane.

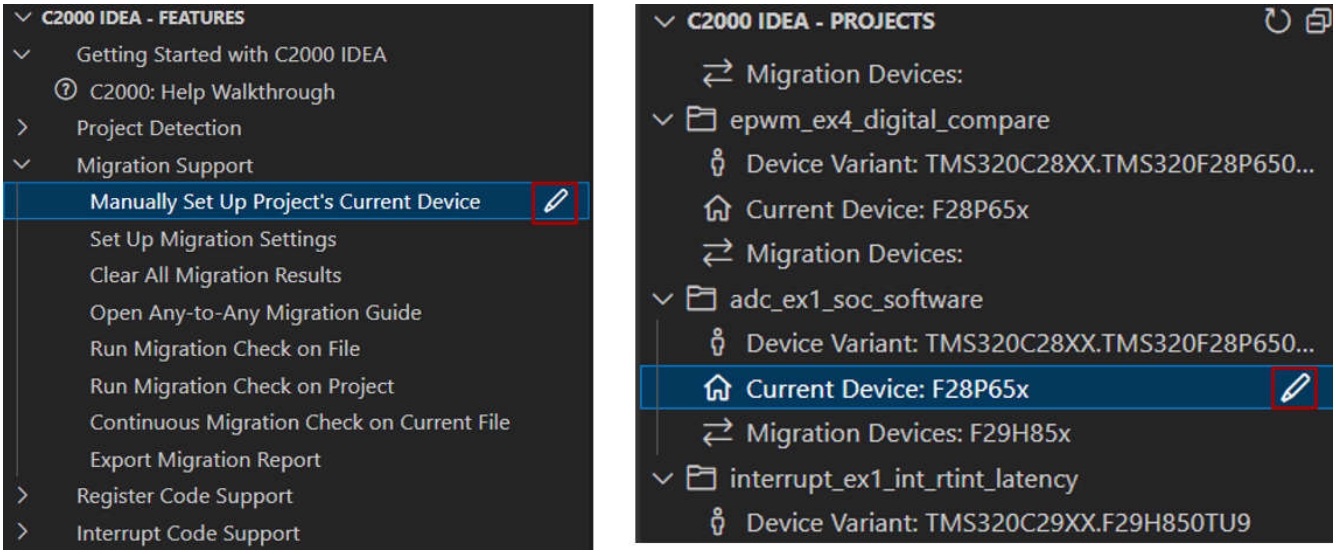

Figure 3-4. Manual Update of Current Device of the Project

This page allows users to customize the migration process by selecting devices and specifying exceptions for files, folders, and code changes.

- **Selecting Migration Devices**:
  - Users can choose one or multiple migration devices by checking the selection box within the Migration Settings page.
- **Excluding Files and Folders**:
  - Specifies files and folders to be ignored during migration in the *Migration Files* and *Folders Exception* section.
  - Entries must be separated using a semicolon (;).
- **Excluding Code Changes**:
  - Specifies code modifications to be ignored during migration in the *Migration Check Exceptions* section.
  - Entries must be separated using a semicolon (;).

After entering the required details, save the migration setup page. The Migration Devices information are updated accordingly in the C2000 IDEA - Projects pane within the extension tree.

3.3 Migration Execution

The C2000 IDEA extension can be used to run a F28x-to-F28x or F28x-to-F29x migration check on a project or file written with driverlib-style-code. This code style is characterized by using calls to functions defined in the driverlib source files (ex: Device_init ()) and/or register accesses containing the _o_syntax.

3.3.1 Running Migration Check on a Standalone File

This extension allows users to perform a migration check on a standalone file. To execute the migration check, follow these steps:

- Open the standalone file and make sure the file is set as the active file in the editor.
- Run the migration check using one of the following methods:
  1. Enter CTRL+SHIFT+P and click *C2000: Run Migration Check on File.*
  2. Extension tree methods:
     a. Navigate to Migration Support > *Run Migration Check on File* in the *C2000 IDEA - Features* pane.
     b. Click the icon at right top of an active file editor as shown in Figure 3-5.

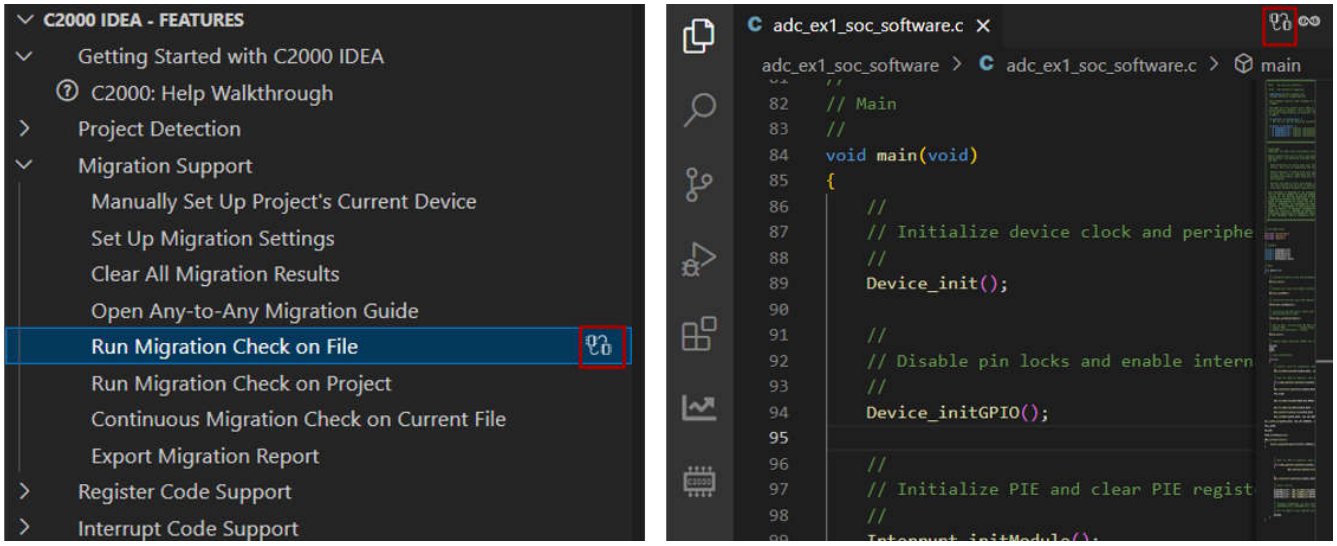

Figure 3-5. File Migration Execution (Extension Tree and Editor View)

After running the migration check on an active file, a progress bar appears at the bottom right corner of the screen. This progress bar provides real-time updates, displaying the file path currently being processed. Once the migration process is complete, the progress bar is replaced with a status message in the same location, indicating *Migration check completed*.

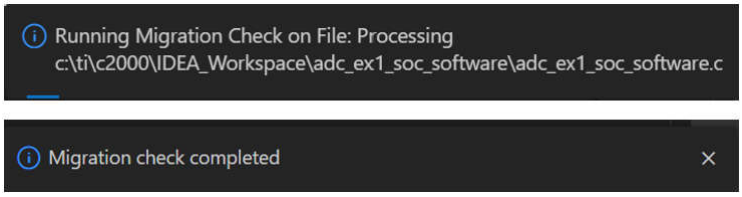

Figure 3-6. File Migration Progress Bar and Completion Notification

Continuous Migration Check on Current File: this feature automatically runs a migration check on the active file at regular intervals whenever changes are detected. This makes sure that users can promptly identify and address migration concerns without manually re-executing the check after each modification, improving efficiency and workflow.

Note

Do not use any migration features when the *C2000: Continuous Migration Check on Current File* feature is enabled in the tool.

- Users can enable the continuous migration check on the active file using one of the following methods:
  1. Enter CTRL+SHIFT+P and click *C2000: Enable Continuous Migration Check on Current File.*
  2. Extension tree methods:
     a. Navigate to Migration Support > *Continuous Migration Check on Current File* in the *C2000 IDEA - Features* pane.
     b. Click the icon at the top right of the active file editor.

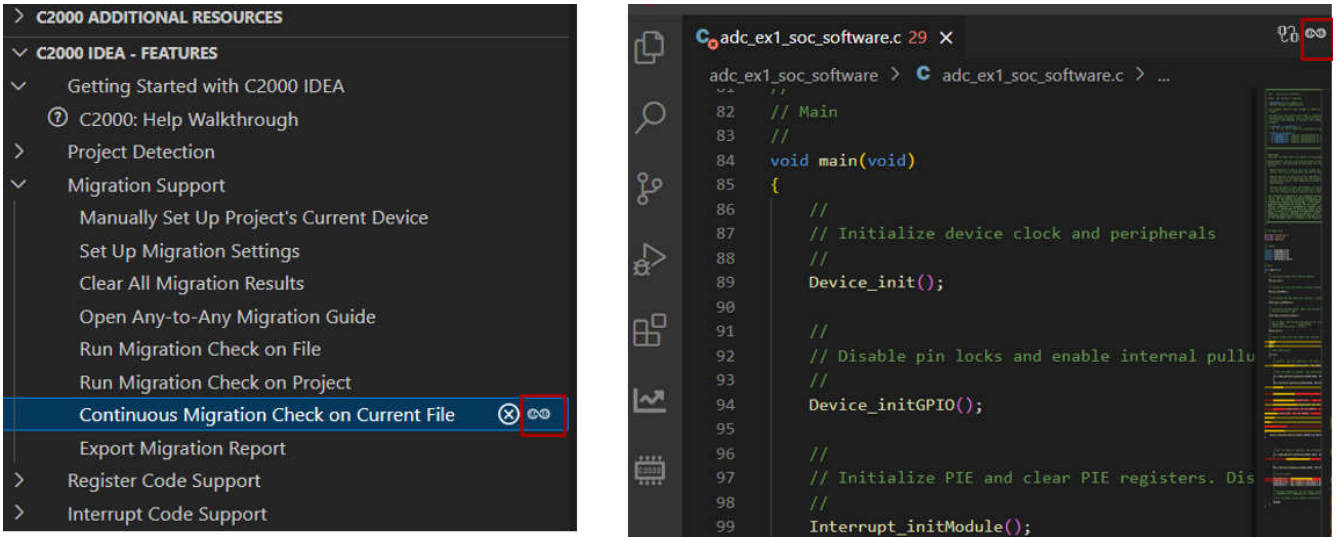

Figure 3-7. Enabling Continuous Migration Check on Current File

- Users can disable the continuous migration check on the active file using one of the following methods:
  1. Enter CTRL+SHIFT+P and click *C2000: Disable Continuous Migration Check on Current File.*
  2. Extension tree methods:
     a. Navigate to Migration Support > *Continuous Migration Check on Current File* in the *C2000 IDEA - Features* pane.
     b. Click the icon at the top right of the active file editor.

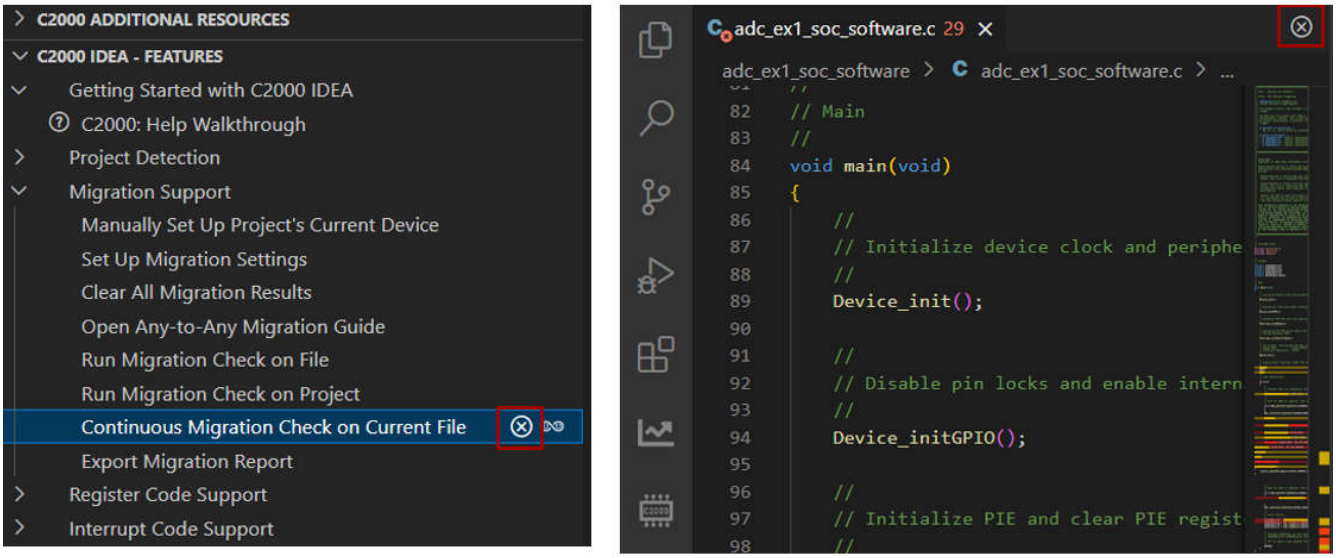

Figure 3-8. Disabling Continuous Migration Check on Current File

3.3.2 Running Migration Check on a Project

This extension allows users to perform a migration check on an entire C2000 device project. To execute the migration check, follow these steps:

- Make sure that the *Migration Settings Setup* page is customized according to the requirements.
- To run the migration check on a project, use one of the following methods:
  1. Enter CTRL+SHIFT+P and click *C2000: Run Migration Check on Project.*
  2. Extension tree methods:
     a. Navigate to Migration Support > *Run Migration Check on Project* in the *C2000 IDEA - Features* pane.
     b. Click the icon as shown in Figure 3-9, next to Migration Devices under the detected project in the *C2000 IDEA - Projects* pane.

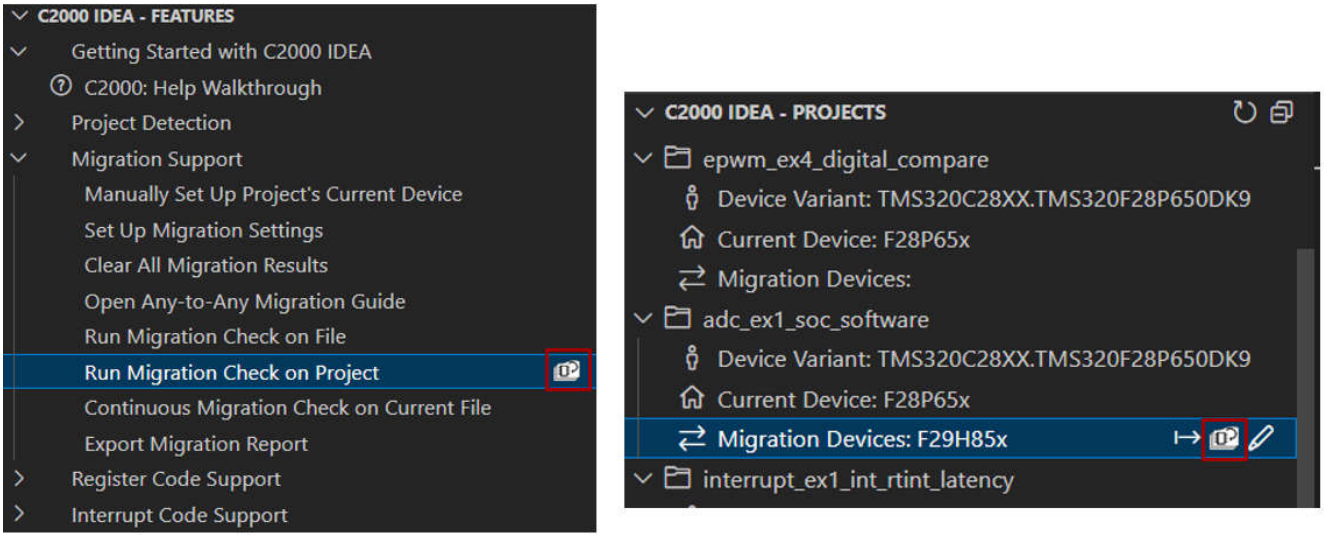

Figure 3-9. Project Migration Execution (Extension Tree)

Note

An error is thrown by the extension if there is any overlap of folders or files in the migration folder and files to be ignored information in the migration setup page.

Note

Do not use these migration features while the tool is already running a migration check on the project. Wait for the *Migration check completed on [project name]* at the bottom right of the screen to show before enabling other migration features. The amount of time taken to run the check entirely depends on how many files, lines and code changes exist in a project. The migration report includes the time taken on each file.

After running the migration check on the entire project, a progress bar with a percentage indicator appears at the bottom right corner of the screen. This progress bar provides real-time updates, displaying the file path currently being processed during the migration.

Once the migration process is complete, the progress bar is replaced with a status message in the same location indicating *Migration check completed on [project name].*

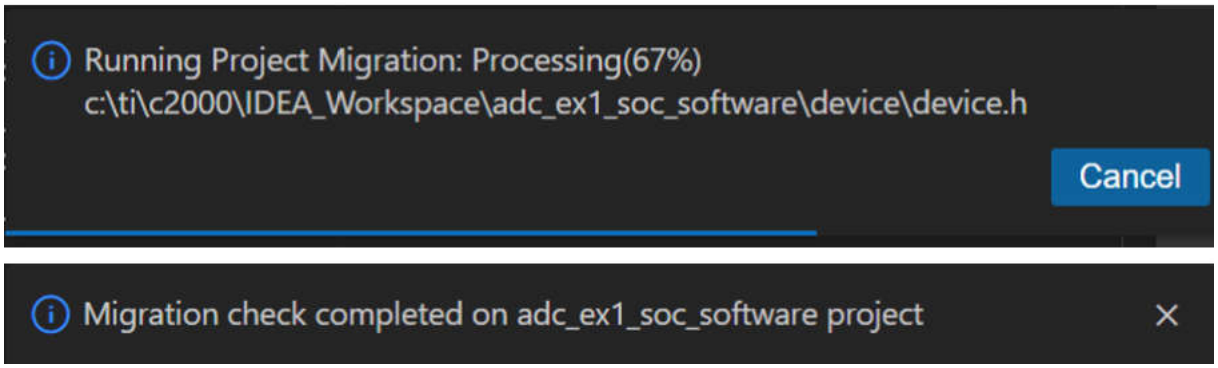

Figure 3-10. Project Migration Progress Bar and Completion Notification

The project migration summary can be viewed in the *Output Console* of Code Composer Studio (CCS). This summary provides information about the migration process, including ignored files and folders.

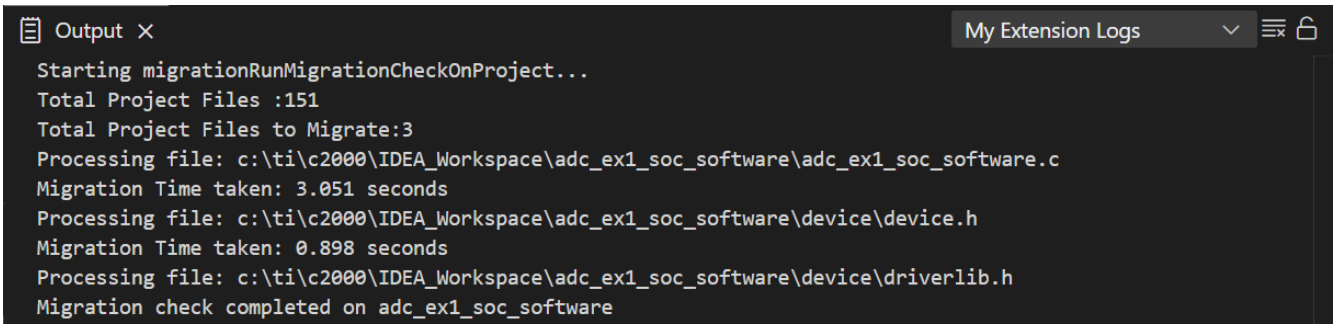

Figure 3-11. CCS Output Console after Project Migration

3.4 Quick Fixes

The C2000 IDEA extension detects and highlights migration concerns when executing a migration run on a standalone file or an entire project. All major migration concerns in the file are underlined with a red squiggly line and all other migration warnings in the file are underlined with a yellow squiggly line.

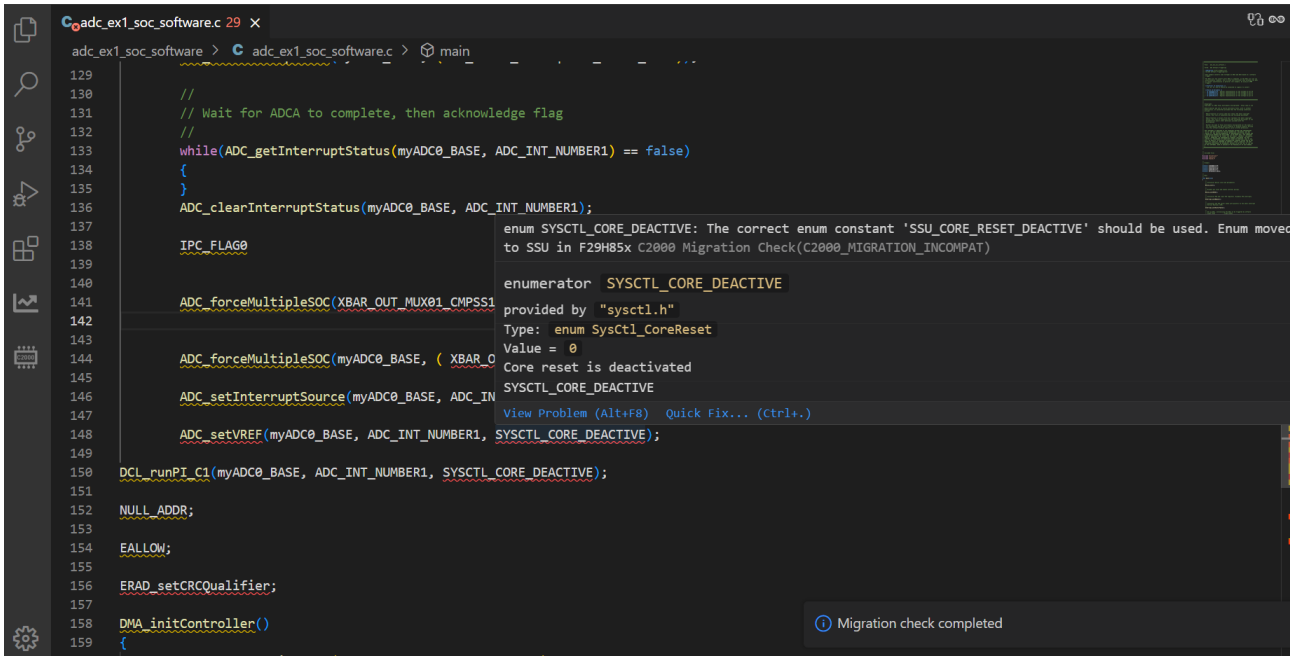

Figure 3-12. File View After Migration Execution

Each highlighted migration concern includes a prompt message, along with *View Problem* and *Quick Fix* options:

- View Problem: Displays a detailed description of the migration concern, explaining the differences between the current and migration device.
- Quick Fix: Suggests available answers to address the migration concern efficiently.

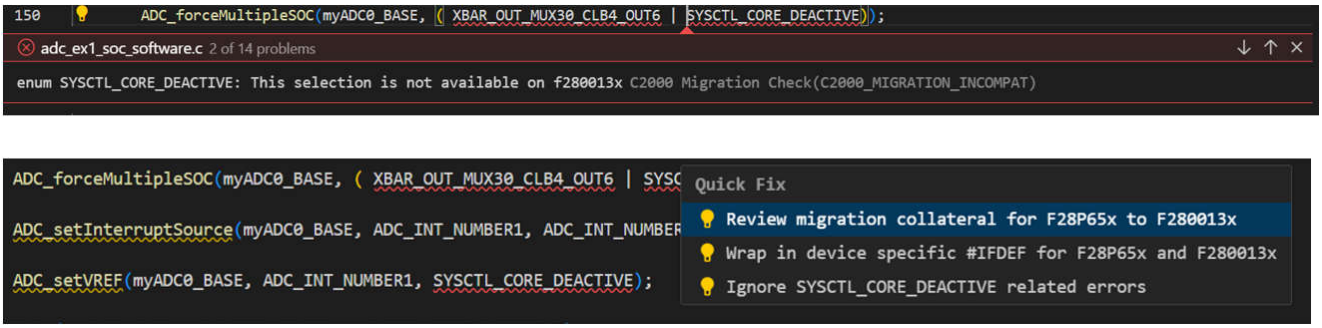

Figure 3-13. Migration Concern - View Problem and Quick Fix

This extension offers several options to help users efficiently resolve migration concerns and make sure of smooth transitions between devices. Users can choose from the following resolutions:

- *Review migration collateral for [current device] to [migration device]*:
  - Opens a link to C2000 Device Migration Report Generation for the latest C2000WARE online migration collateral, providing detailed guidance for the specific migration code change.

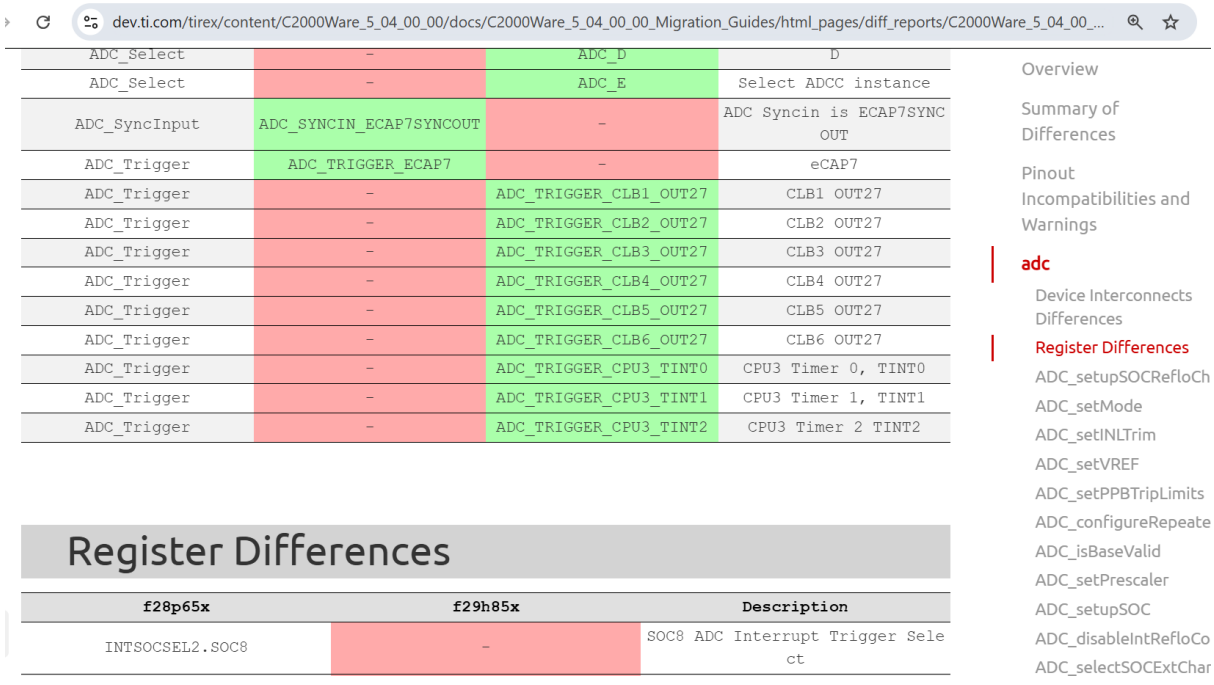

Figure 3-14. Migration Collateral HTML Page

- *Wrap in device specific #IFDEF for [current device] and [migration device]*
  - Automatically generates preprocessor conditionals (#IFDEF) around the line of code so that an updated version of the code can be compiled for the new device and add a #define for the current device somewhere in the file.
  - Allows users to define an alternate code implementation tailored for the new device. Inserts a placeholder comment (// Enter alternate code) where modifications need to be made.
  - This extension has capability to suggest code replacement for most of F28x-to-F29x migration concern.

```
Start of device specific migration code - F28P65x
#if F28P65x //_DEVICE_MIGRATION_
        ADC_forceMultipleSOC(XBAR_OUT_MUX01_CMPSS1_CTRIPOUTL, ( XBAR_OUT_MUX30_CLB4_OUT6 | SYSCTL_CORE_DEACTIVE));
Change of device specific migration code - F280013x
#elif F280013x //_DEVICE_MIGRATION_
        // Enter alternate code
End of device specific migration code - F28P65x/F280013x
#endif //_DEVICE_MIGRATION_
```

Figure 3-15. Wrap #IFDEF Quick Fix (F28x-to-F28x)

```
Start of device specific migration code - F28P65x
#if F28P65x //_DEVICE_MIGRATION_
        ADC_forceMultipleSOC1(XBAR_OUT_MUX01_CMPSS1_CTRIPOUTL, ( XBAR_OUT_MUX30_CLB4_OUT6 | SYSCTL_CORE_DEACTIVE));
Change of device specific migration code - F29H85x
#elif F29H85x //_DEVICE_MIGRATION_
        // Suggested replacement: ADC_forceMultipleSOC1(XBAR_OUT_MUX01_CMPSS1_CTRIPOUTL, ( XBAR_OUT_CLB4_OUT6 | SYSCTL_CORE_DEACTIVE));
End of device specific migration code - F28P65x/F29H85x
#endif //_DEVICE_MIGRATION_
```

Figure 3-16. Wrap #IFDEF Quick Fix (F28x-to-F29x)

- *Ignore code related errors*
  - Suppresses the migration concern by adding to the Migration Check Exceptions section in the *Setup Migration Settings* page.
  - Typically used when user have implemented a different approach to handle the migration concern manually.
- *All Enum fixes Wrap in device specific #IFDEF for [current device] and [migration device]*
  - Functions similarly to the #IFDEF prior quick fix but is specifically designed for enumeration-based migration concerns.
  - Appears only when migrating to F29H85x and if multiple enum-related changes exist on the same line.
  - Automatically applies #IFDEF wrappers for all relevant enum migration concerns in that line.

```
Start of device specific migration code - F28P65x
#if F28P65x //_DEVICE_MIGRATION_
        ADC_forceMultipleSOC1(XBAR_OUT_MUX01_CMPSS1_CTRIPOUTL, ( XBAR_OUT_MUX30_CLB4_OUT6 | SYSCTL_CORE_DEACTIVE));
Change of device specific migration code - F29H85x
#elif F29H85x //_DEVICE_MIGRATION_
        // Suggested replacement:ADC_forceMultipleSOC1(XBAR_OUT_CMPSS1_CTRIPOUTL, ( XBAR_OUT_CLB4_OUT6 | SSU_CORE_RESET_DEACTIVE));
End of device specific migration code - F28P65x/F29H85x
#endif //_DEVICE_MIGRATION_
```

Figure 3-17. All Enum Wrap #IFDEF Quick Fix (F28x-to-F29x)

By leveraging these quick fix options, users can significantly reduce manual effort, simplify the migration process, and make sure iof code compatibility across devices.

3.5 Migration Report

Migration Report plays a crucial role in analyzing code changes when transitioning between C2000 devices. This provides a structured overview of modifications, helping users assess migration complexity, identify potential issues, and select the best target device. With detailed diagnostics and export capabilities, Migration Report make sure of a simplified and efficient migration process.

This report can be generated after executing the migration, either for the entire project or a specific active file. This flexibility allows developers to focus on critical sections while maintaining a comprehensive overview of the migration impact.

There are multiple ways to generate and export the migration report:

1. Enter CTRL+SHIFT+P and click *C2000: Export Migration Report.*
2. Extension tree methods:
   a. Navigate to Migration Support > *Export Migration Report* in the *C2000 IDEA - Features* pane.
   b. Click the icon as shown in Figure 3-18, next to Migration Devices under the detected project in the *C2000 IDEA - Projects* pane.

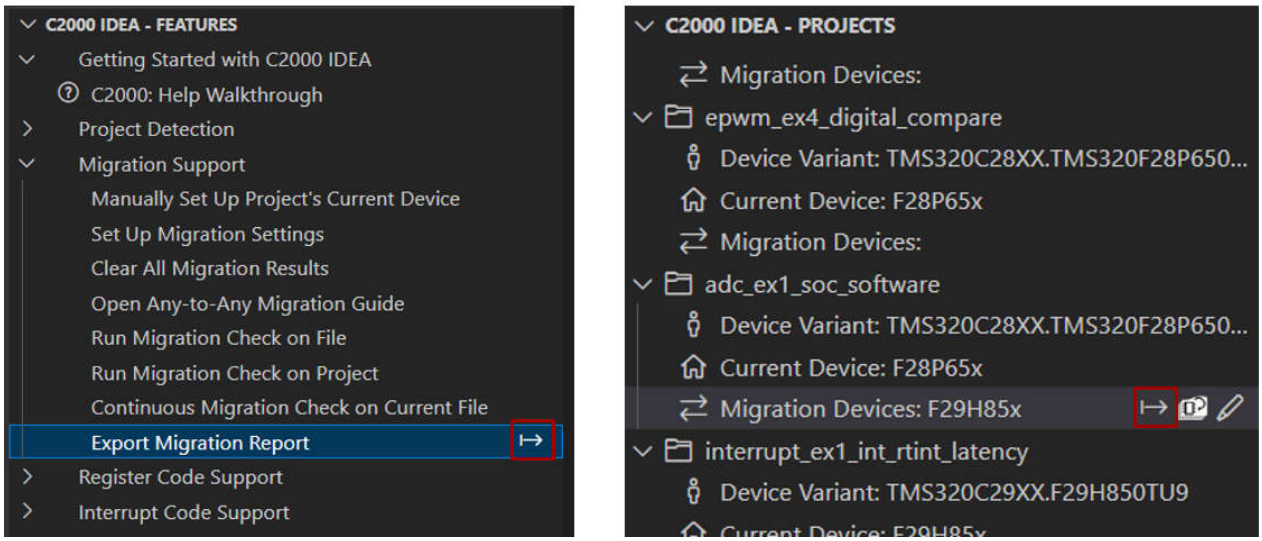

Figure 3-18. Export Migration Report (Extension Tree)

Once generated, the report can be exported and saved to a desired location using the file browser, verifying that migration data is readily available for further analysis and documentation.

Key Sections in Migration Report

1. Migration Info - provides an overview of the migration process
   a. *Source and Target Devices*: displays the original device and the selected migration device or devices.
   b. *Ignored Symbols, Files & Folders*: lists excluded symbols, files, and directories that were not considered during migration.
   c. *Processed Files*: specifies the analyzed files that underwent migration.
   d. *Migration Time*: indicates the time taken for each file's migration, offering insights into performance and complexity.
2. *Migration Diagnostics* - highlights detected changes and potential issues in the migration process
   a. *Classification*: each detected change is categorized as a warning or an error, based on severity.
   b. *Detailed Code Change Analysis*: provides a clear explanation of the differences between the source and target device.
   c. *Location*: specifies the exact Line (Ln) and Column (Col) of the affected code, making issue resolution more efficient.

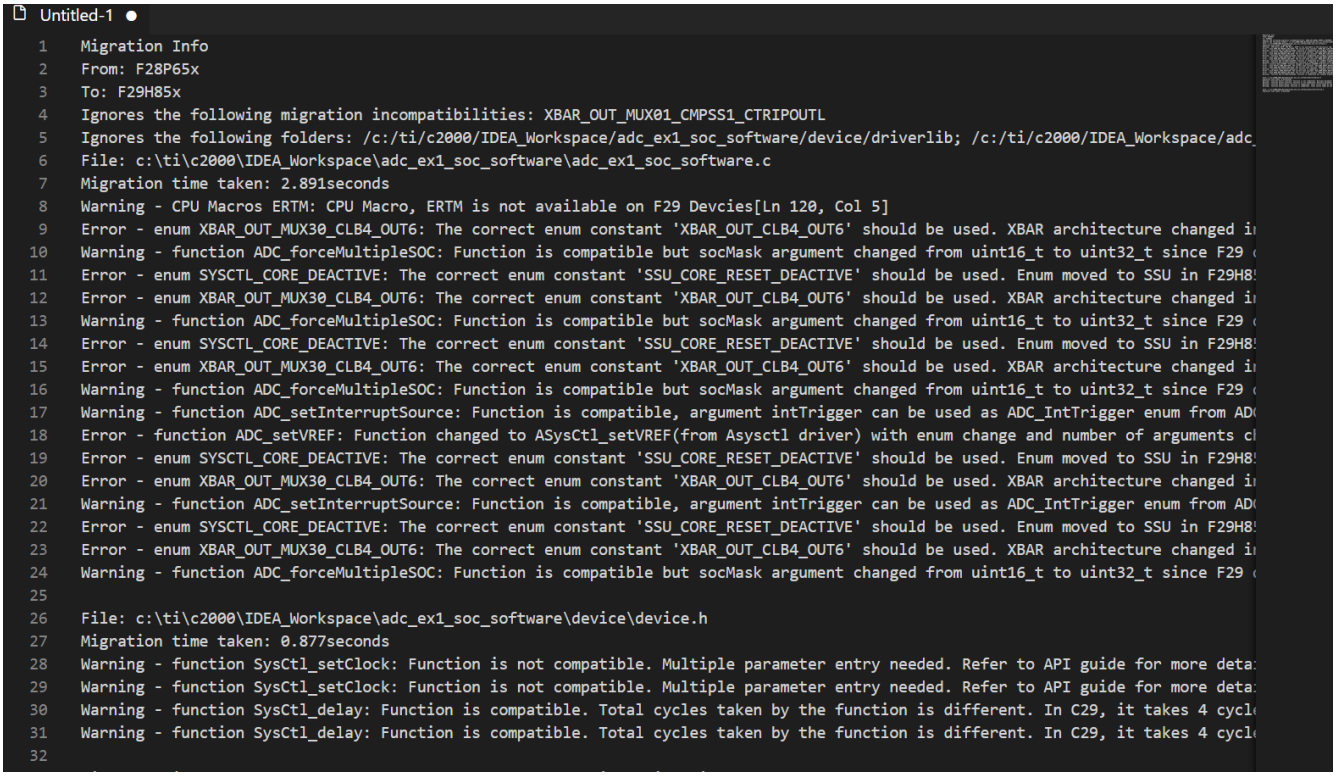

Figure 3-19. Migration Report

By exporting the report and saving for documentation, developers can reference migration details whenever needed. The report is particularly useful for analyzing migration complexity across multiple target devices, helping developers assess potential challenges and determine the best approach for migration. With clear insights into code modifications and potential roadblocks, developers can make informed decisions about selecting an excellent migration device.

With structured diagnostics, export capabilities, and detailed insights, the Migration Report is an indispensable resource for verifying a seamless and efficient transition between C2000 devices.

3.6 Bitfield Migration

The C2000 IDEA Extension can be used to run a F28x-to-F28x or F28x-to-F29x migration check on a file written with bitfield-style code. This code style is characterized by using calls to functions defined in the bitfield source files (ex: InitSysCtrl()) and/or register accesses containing the [base name].[register name].all or [base name].[register name].bit.[field name] syntax.

Follow these steps to enable Bitfield Migration:

1. Open a C2000 application C-Code file.
2. Run the migration check by pressing CTRL+SHIFT+P, typing and selecting *C2000: Run Bitfield Migration Check on File*
3. Select the current C2000 device the code in the file applies to.
4. Select the C2000 device to migrate the file to.
5. The status bar at the bottom right of the screen displays *Finished Bitfield Migration from [current device] to [migration device]* when finished. All migration concerns in the file are underlined with a red squiggly line.
6. Review and resolve concerns throughout the file. The following options are provided when the underlined code is hovered over:
   a. Select *View Problem* to quickly loop through the detected concerns in the file.
   b. Select *Quick Fix* to mitigate the migration concern. Select one of the below options:
      i. *Review migration collateral for [current device] to [migration device]*- This option opens a link to the online migration collateral for the specific migration path using the latest version of C2000WARE.

ii. *Wrap in device specific #IFDEF for [current device] and [migration device]* - This option autogenerates pre-processor wrappers around the line of code so that an updated version of the code can be compiled for the new device. Fill in the line with the *//Enter alternate code* comment with the modified code and add a #define for the current device somewhere in the file.
iii. *Ignore code related errors* - This option ignores this migration concern.

4 Summary

The Texas Instruments' C2000 SDKs provide essential tools, including drivers and libraries, to simplify real-time control application development. With features like C2000 SysConfig for simplified code initialization and peripheral configuration, the ecosystem is designed to support developers at all levels. Building on this foundation, the C2000 IDEA tool centralizes the development environment, making coding and debugging more efficient. The C2000 IDE Assist Extension (IDEA) further enhances this by facilitating seamless migration across device portfolios and transitioning legacy code to new devices.

As a powerful extension for Code Composer Studio (CCS) 20, IDEA accelerates software migration by automatically analyzing customer code, identifying potential migration issues across TI MCUs, and providing optimized resolutions with contextual documentation. The tool generates a migration report, offering a structured analysis of code modifications, highlighting compatibility concerns, and assisting in selecting the best target device. Designed for both beginners and experienced developers, IDEA enhances productivity by simplifying migration tasks, making sure of a smooth transition between devices. By integrating live assistance at every stage of development, the IDEA Tool significantly reduces the time and effort required for embedded software migration. The intelligent automation, and simplified interface make IDEA Tool the first resolution in the embedded software world capable of enabling efficient, high-performance development for real-time microcontrollers.

5 References

Tools and Software:

- Texas Instruments, *C2000 IDEA Open VSX (VSIX Download)*
- Texas Instruments, *C2000 IDEA GitHub Repository (VSIX Download)*
- Texas Instruments, *Code Composer Studio (CCS) IDE*
- Texas Instruments, *C2000WARE (F28x SDK)*
- Texas Instruments, *F29X-SDK (F29x SDK)*

Documentation:

- Texas Instruments, *C28x Academy - Migration Resources*
- Texas Instruments, *C29x Academy - Migration Resources*
- Texas Instruments, *F28x to F29x Software Migration Guide*
- Texas Instruments, *Application Software Migration to the C29 CPU Application Note*
- Texas Instruments, *TMS320F2837x, TMS320F2838x, TMS320F28P65x Migration to TMS320F29H85x User's Guide*
- Texas Instruments, *C2000 Design & Development*

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265