Using Visual Studio Code

# Roundtable TSMS

Roundtable Development Team
Updated October 2026



# Table of Contents

1 Overview ..... 1
2 Analysis ..... 1
3 Implementation ..... 2
3.1 Event Hook Change Summary ..... 2
3.2 The Code ..... 3
4 Conclusion ..... 4

# 1 Overview

With the availability of the OpenEdge ABL extension and language server from Riverside Software, Visual Studio Code has become a popular development environment for OpenEdge ABL development. While Roundtable TSMS does not currently provide native integration, developers can successfully use Visual Studio Code when working with code that is checked out of Roundtable with a Task-share or Lab status. By utilizing the Workspace and Task information from the Roundtable repository, the necessary data can be used to facilitate the automatic creation of the configuration files required by the ABL extension and Visual Studio Code. This approach enables organizations to adopt modern development tooling while continuing to follow established Roundtable TSMS source control and task management processes.

# 2 Analysis

Visual Studio Code support in a Roundtable TSMS environment is typically based on two configuration artifacts: an `openedge-project.json` file for the OpenEdge ABL extension and, optionally, a `.code-workspace` file for workspace organization and navigation.

To use the ABL extension, an `openedge-project.json` configuration file must exist in the Visual Studio Code project folder. This configuration defines critical development settings such as source directories, build-paths, PROPATH entries, database connections, and other environment-specific data that is required for proper language services and compilation support.

Although optional, a Visual Studio Code `.code-workspace` file can significantly enhance the development experience. The workspace file can be configured to include both the developer's Task directory and a read-only reference to the central Roundtable Workspace. This provides developers and AI-assisted coding tools with visibility into both the active task changes and the baseline application code, improving code navigation, dependency discovery, and contextual analysis.

Because the information required to generate these files already exists within Roundtable TSMS, manual creation introduces unnecessary effort and increases the risk of configuration errors. To address this, Roundtable TSMS includes a utility beginning with version 12.8.3 that can automatically generate the required Visual Studio Code configuration artifacts based on the selected Task and Workspace context. The utility, `rtb_createvscodeconfig.p`, is distributed as open source within the `rtb/util/p` directory of the Roundtable installation. It can be executed manually or invoked automatically through the Roundtable `CHANGE-TASK` event hook, ensuring project configuration remains synchronized with the developer's current task context.

The example code in the following section shows how this can be accomplished.

# 3 Implementation

You can automate the creation of the previously mentioned configuration files by customizing the appropriate Roundtable event hook procedure.

- For a client-server environment, modify **rtb_events.p**.
- For an AppServer environment, modify **rtb_events_cl.p**.

Both programs can be found in the root of your Roundtable installation.

The goal for the event hook customization is to make switching to a Roundtable task seamless for developers by:

- Creating OpenEdge VS Code project configuration files on demand
- Eliminating manual setup steps
- Automatically opening the task in VS Code

Implementing these changes turns Task selection into a "ready-to-code" experience with little or no manual intervention.

## 3.1 Event Hook Change Summary

For this customization, we are going to intercept the `CHANGE-TASK` event. This event fires after a Task selection is made in Roundtable.

We first determine whether the task is eligible for Visual Studio Code support by verifying that it is not a Central-share task. We then check if an `openedge-project.json` configuration file already exists in the task directory and prompt the user to either create or replace it.

If the user agrees, we call `rtb_createvscodeconfig.p` to generate VS Code and OpenEdge configuration files for the selected Task.

We then optionally attempt to launch VS Code and open the task directory, reusing an existing VS Code window if possible.

### 3.2 The Code

```
CASE pcEvent:

  WHEN {&CHANGE-TASK} THEN DO:

    DEFINE VARIABLE cShare AS CHARACTER NO-UNDO.
    DEFINE VARIABLE cDir   AS CHARACTER NO-UNDO.
    DEFINE VARIABLE cCmd   AS CHARACTER NO-UNDO.
    DEFINE VARIABLE iTask  AS INTEGER   NO-UNDO.
    DEFINE VARIABLE iRes   AS INTEGER   NO-UNDO.
    DEFINE VARIABLE lOk    AS LOGICAL   NO-UNDO.
    DEFINE VARIABLE lExist AS LOGICAL   NO-UNDO.

    ASSIGN
      iTask  = INTEGER(ENTRY(1,pcOther))
      cShare = ENTRY(2,pcOther)
      cDir   = ENTRY(3,pcOther).

    IF iTask > 0 AND cShare <> "Central" THEN DO:

      FILE-INFO:FILE-NAME = cDir + "/" + "openedge-project.json".

      lExist = (FILE-INFO:FULL-PATHNAME <> ?).

      IF NOT(lExist) THEN
        MESSAGE
          "Create ABL configuration file for VS Code project?"
          VIEW-AS ALERT-BOX QUESTION BUTTONS YES-NO UPDATE lOk.
      ELSE
        MESSAGE
          "VS Code ABL configuration file exists. Overwrite?"
          VIEW-AS ALERT-BOX QUESTION BUTTONS YES-NO UPDATE lOk.

      IF lOk THEN DO:

        /*
         Pass task number, optional project name, true/false to overwrite
        */

        RUN rtb/util/p/rtb_createvscodeconfig.p (iTask, "", TRUE, OUTPUT cError).

        SESSION:SET-WAIT-STATE("").

        IF cError <> "" THEN
          MESSAGE cError VIEW-AS ALERT-BOX ERROR.

      END.

      /* Optional - automatically run VSCode */

      lExist = (FILE-INFO:FULL-PATHNAME <> ?).

      IF lExist THEN DO:

        /* Open the code workspace.. */
        cCmd = OS-GETENV("USERPROFILE") + "\AppData\Local\Programs\Microsoft VS
Code\code " + REPLACE(cDir,"/","~\") + "\task.code-workspace -r".

        /*.. or open the task folder only */
```

```
        /*cCmd = OS-GETENV("USERPROFILE") + "\AppData\Local\Programs\Microsoft VS
Code\code " + REPLACE(cDir,"/","~\") + " -r".*/

        iRes = DYNAMIC-FUNCTION('CreateProcess', cCmd, "", 1).

        IF iRes = 0 THEN
          MESSAGE
            SUBSTITUTE("Unable to run command '&1'.", cCmd)
            VIEW-AS ALERT-BOX ERROR.

      END.

    END.

  END CASE.
```

In this example, the utility is invoked with overwrite enabled after user confirmation. Organizations may choose to modify this behavior to align with their preferred workflow or configuration management practices.

# 4 Conclusion

Automating the Visual Studio Code configuration file through Roundtable TSMS provides a practical bridge between established OpenEdge ABL development processes and modern tooling. By invoking `rtb_createvscodeconfig.p` from the `CHANGE-TASK` event, teams can keep project settings synchronized with the active Task and Workspace, minimize manual setup and configuration errors, and give developers consistent access to both task-specific changes and baseline application code. This streamlined approach maintains productivity while preserving the trusted governance and control provided by Roundtable TSMS.