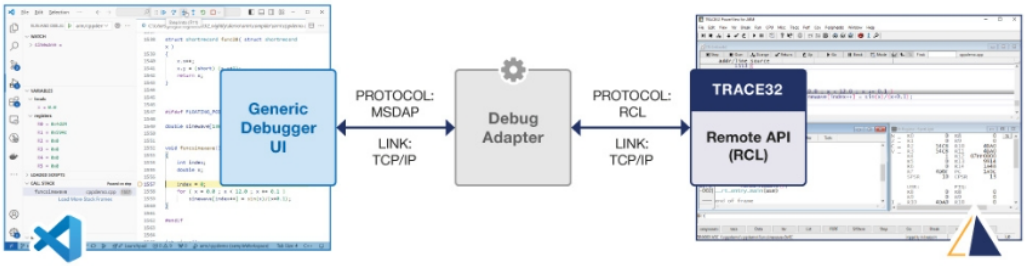
How to use VSCode as Front-End for TRACE32

2026-10-26 - Comments (33) - TRACE32 PowerView

To use Visual Studio Code (VSCode) as a frontend for TRACE32, you can utilize the Debug Adapter Protocol (DAP). Although TRACE32 does not natively support DAP, you can set up a proxy to translate between DAP and TRACE32's Remote API (RCL). This proxy enables VSCode to function as an effective Front-End for TRACE32, facilitating enhanced debugging capabilities.



Usage

You need TRACE32 version Nightly from 2024-07-16 (Build 170887 or later) or starting from the DVD R.2024.09.

Starting from t32debugadapter v 0.0.22, you can use the **launch configuration** to start TRACE32.

Launch Configuration (Starting TRACE32 from VSCode)

1- Open VSCode

2- Add the **launch configuration** in VSCode (Menu → Run → Add Configuration → select empty)

```
{

        "configurations": [
            {
                "type": "node",
                "request": "launch",
                "name": "TRACE32 Launch",
                "debugServer": 58870,
                "trace32Port": 20000,
                "trace32SystemPath": "C:/T32",
                "trace32Target": "t32marm",
                //"screen":false,
                "trace32ConnectionScript": "C:/T32/Simulator.cmm",
                "trace32SetupScript": "~~/demo/arm/compiler/gnu/demo_sram.cmm",
                "trace32SetupScriptTimeout": 10,
                "trace32SetupScriptArgs": [
                    "arg1",
```

```
                    "arg2"
                ]
            }
        ]
}
```

3- Run **t32debugadapter.exe** in cmd line (can be found under ~~/demo/env/vscode), or check the Tip below

4- In VSCode, navigate to **Run and Debug** and start a debug session with "**TRACE32 Launch**".

Attach Configuration (Connecting to a running TRACE32)

1- Start TRACE32 with API enabled:

- By adding this block to the config file:

```
RCL=NETTCP
PORT=20000
```

- Or using the command line option (TRACE32 version Nightly from 2024-07-19 (Build 171009 or later)):

```
--t32-api-rcl=TCP 20000
```

Example: t32marm.exe --t32-api-rcl=TCP 20000 -c config.t32

- Or using the PRACTICE commands (TRACE32 version Nightly from 2024-07-19 (Build 170994 or later)):

```
SETUP.API.RCL /Interface TCP /Port 20000.
```

2- Add the **Attach configuration** in VSCode

```
{
        "configurations": [
            {
                "type": "node",
                "request": "attach",
                "name": "TRACE32 Attach",
                "debugServer": 58870,
                "trace32Port": 20000,
                "trace32Node": "localhost",
                "trace32SetupScript": "~~/demo/arm/compiler/gnu/cppdemo.cmm",
                "trace32SetupScriptTimeout": 10,
                "trace32SetupScriptArgs": [
                    "arg1",
                    "arg2"
                ]
            }
        ]
}
```

3- Run **t32debugadapter.exe** in cmd line (can be found under ~~/demo/env/vscode), or check the Tip below

4- In VSCode, navigate to **Run and Debug** and start a debug session with "**TRACE32 Attach".**

Tip:

You can automatically start the debug adapter (step 3) by using a preLaunchTask:

launch.json:

```
{
        "configurations": [
            {
                "type": "node",
                "request": "launch",
                "name": "TRACE32 Launch",
                "debugServer": 58870,
                "trace32Port": 20000,
                "preLaunchTask": "start_debug_adapter",
                "trace32SystemPath": "C:/T32",
                "trace32Target": "t32marm",
                                //"screen":false,
                "trace32ConnectionScript": "C:/T32/Simulator.cmm",
                "trace32SetupScript": "~~/demo/arm/compiler/gnu/demo_sram.cmm",
                "trace32SetupScriptArgs": [
                    "arg1",
                    "arg2"
                ]
            },
                        {
                "type": "node",
                "request": "attach",
                "name": "TRACE32 Attach",
                "preLaunchTask": "start_debug_adapter",
                "debugServer": 58870,
                "trace32Port": 20000,
                "trace32Node": "localhost",
                "trace32SetupScript": "~~/demo/arm/compiler/gnu/cppdemo.cmm",
                "trace32SetupScriptArgs": [
                    "arg1",
                    "arg2"
                ]
            }
        ]
}
```

and tasks.json:

```
{
    "version": "2.0.0",
    "tasks": [{
        "label": "start_debug_adapter",
        "command": "start C:/T32/demo/env/vscode/bin/windows64/t32debugadapter.exe",
// adjust path as needed
        "args": [],
        "type": "shell"
    }]
}
```

Tip:

If you have problem using the default port 58870, you can start t32debugadapter with the parameter:

```
t32debugadpter.exe --port=50000
```

Of course, make sure to set this port value to the “debugServer” parameter in the VSCode configuration.

Tip:

Using t32debugadapter version 0.0.20, you can start VSCode and t32debugadapter on a machine like WSL, and TRACE32 on windows, and use the parameters “translateInputPath” and “translateOutputPath” like below:

```
{
    "configurations": [
        {
            "type": "node",
                "request": "attach",
                "name": "dapserver attach",
                //"preLaunchTask": "start_debug_adapter",
                "debugServer": 58870,
                "trace32SetupScript": "~~/demo/arm/compiler/gnu/cppdemo.cmm",
                "trace32SetupScriptArgs": [
                    "arg1",
                    "arg2"
                ]
                "trace32Node": "172.21.224.1", //IP address
                "trace32Port": "20000", //API port
                "translateInputPath": [["C:\\T32\\demo\\arm\\compiler\\gnu\\src",
"\\mnt\\c\\T32\\demo\\arm\\compiler\\gnu\\src"]],
                "translateOutputPath":
[["\\mnt\\c\\T32\\demo\\arm\\compiler\\gnu\\src",
"C:\\T32\\demo\\arm\\compiler\\gnu\\src"]]

        }
    ]
}
```

The API port on TRACE32 needs to be launched with the option `/RemoteHost ALLOW`

```
SETUP.API.RCL /Interface TCP /Port 20000. /RemoteHost ALLOW
```

Comments (33)

Comments (33)

SB **Sepehr Bijani**
2 years ago
Hello, Since we are using linux environment for debugging and build, could you provide "t32debugadapter" for linux? BR, Sepehr Bijani
Wiem Wala Benayed
2 years ago

The Linux version will be available soon in our website.

For the moment, I will send it via email.

FLAVIO VINCENZO CONDEMI
2 years ago
Can I have the t32debugadapter linux version too?
Wiem Wala Benayed
2 years ago

The Linux and Mac versions are available in our website

FLAVIO VINCENZO CONDEMI
2 years ago
I just checked and I see there is a file in bin/pc_linux64 folder called t32debugadapter. In windows I execute the .exe file (I find it in bin/windows folder). I'm wondering how I can execute this file in Linux. I tried with "source t32debugadapter" command, but I got an error. Do I need to copy and paste the file somewhere else? Thank you in advance
Wiem Wala Benayed
2 years ago

You can simply call it in a terminal with: " <PATH>/t32debugadapter"

Kyrylo M
2 years ago
Tried to use this configuration and it fails on launch with following error: InitializeRequestArguments.__init__() got an unexpected keyword argument supportsANSIStyling' My Trace32 version is: TRACE32 for ARM Nightly Build (64-bit) Software Version: N.2024.10.000173431 Build: 173431. I got t32debugadapter.exe from https://repo.lauterbach.com/scripts/misc/env~vscode_/misc-env-vscode_20240921000808_all_files.zip
Wiem Wala Benayed
2 years ago

The error 'InitializeRequestArguments.__init__() got an unexpected keyword argument supportsANSIStyling' is now fixed.
Please use the new version (v0.0.7) of t32debugadapter available in our website.

Bo Mellberg
2 years ago
Hi! I have the following setup: Linux build system, with Lauterbach setup via USB. Debugging works fine using the Trace32 front-end on Ubuntu. VSCode frontend on Windows laptop, connected to the Linux build system using Remote Host over SSH. This allows me to perform developing tasks easily from home as well as in-office, but the debugging still needs to be done on Ubuntu front-end (via Remote Desktop). How can I use VSCode as a debug front-end on my setup? All I can come up with is running the t32tcpusb on the Linux machine and on top of that run the t32debugadapter.exe on the Windows machine.
Wiem Wala Benayed
2 years ago

You can use the parameters "trace32Node" to set the IP address of the Linux machine. In this case TRACE32 can be started in Linux and VSCode + t32debugadapter in windows.

In this case, you may not see the source file as it is located in the Linux machine.

Bo Mellberg
1 year ago
I get "ECONNREFUSED 127.0.0.1:58870" when I try that. { "type": "node", "request": "attach", "name": "dapserver attach", "debugServer": 58870, "trace32Node": "10.142.0.139", "trace32Port": 20000 } trace32 is running on the Linux host with RCL=NETTCP and PORT=20000
Wiem Wala Benayed
1 year ago

Please make sure to use v0.0.8 of t32debugadapter

Bo Mellberg
1 year ago
"Please make sure to use v0.0.8 of t32debugadapter" Where can I find this? I can't even find the old version 0.0.7 anymore.
Kristoffer Martinsson
1 year ago
https://repo.lauterbach.com/scripts.html Search for vscode
P.Z.
1 year ago
Is there a way to instruct trace32 from VSC to do CD.DO instead of DO only, when using an optional cmm to be loaded? I know I can put CD.DO inside the cmm script but this is not the most convenient way for me. We are

running this in a specific system where the debugger HW is on Windows side while the VSC is inside WSL and docker container. Also the path translation option would be nice to have in a future version of t32debugadapter,
Wiem Wala Benayed
1 year ago

Sorry, there is no just option.

You can just include CD <path> in the beginning of your cmm script.
In the VSCode configuration, the path of the cmm must be windows style, as it will be sent via API to the TRACE32 PowerView.

AT **Alessandro Turano**
1 year ago
I get this error with RCL debug adapter v0.0.8, but I can't find any info about it online:
C:\T32\demo\env\vscode\bin\windows64>t32debugadapter.exe 2025-05-21 16:27:45,014 - lauterbach.msdap - INFO - Lauterbach MSDAP -> RCL debug adapter v0.0.8 2025-05-21 16:27:47,182 - lauterbach.msdap - INFO - request, initialize 2025-05-21 16:27:47,182 - lauterbach.msdap - INFO - response, initialize 2025-05-21 16:27:47,182 - lauterbach.msdap - INFO - event, initialized 2025-05-21 16:27:47,182 - lauterbach.msdap - INFO - request, attach 2025-05-21 16:27:47,243 - lauterbach.msdap - INFO - response, attach 2025-05-21 16:27:47,243 - lauterbach.msdap - INFO - request, setFunctionBreakpoints 2025-05-21 16:27:47,243 - lauterbach.msdap - INFO - response, setFunctionBreakpoints 2025-05-21 16:27:47,247 - lauterbach.msdap - INFO - request, configurationDone 2025-05-21 16:27:47,248 - lauterbach.msdap - INFO - response, configurationDone 2025-05-21 16:27:47,249 - lauterbach.msdap - INFO - request, enableNetworking 2025-05-21 16:27:47,249 - lauterbach.msdap - INFO - response, stopped 2025-05-21 16:27:47,249 - lauterbach.msdap - ERROR - system down 2025-05-21 16:27:47,249 - lauterbach.msdap - INFO - response, enableNetworking 2025-05-21 16:27:47,250 - lauterbach.msdap - INFO - event, stopped 2025-05-21 16:27:47,250 - lauterbach.msdap - ERROR - module '__main__' has no attribute 'EnableNetworkingRequest'
Wiem Wala Benayed
1 year ago

A new version of t32debugadapter v 0.0.11 is now available in our website.

Juhun Choi
1 year ago
Is there a guide to connect multiple Simulators/Cores to the same vscode?
Wiem Wala Benayed
1 year ago

You can add another configuration with different API port value, and before connecting, you select the one that you need to connect to from the list of **RUN AND DEBUG**.

Juhun Choi
1 year ago
Hello, I have another question about CMM, Is there a way to run CMM on VSCode and check result at VSCode? It would be a big help, since our team uses a lot of CMM Scripts for quick debugging. Thanks
Wiem Wala Benayed
1 year ago

It depends on the type of result. It should be something that VSCode can display. For example, the value of a variable.

Could you clarify what specific result you're trying to obtain?

P.Z.
1 year ago
We gave it a try and with the new version of t32debugAdapter (0.0.11) - it seems to very often hang the trace32 application and it is working very slow, This is was not the case for 0.0.8 - is there anyone experiencing the same?
AK **Alan K.**
1 year ago
I have the same feeling. Version 0.0.8 worked more stable. With newest 0.0.11 whole setup becomes unresponsive very often. It works little better when you collapse VARIABLES preview in VSC but still likes to

hang from unknown reason. At the same time, Trace32 enters some strange state requiring to force close it using task manager.

PK **Peter Kunakh**
1 year ago
I have the same. 0.0.8 doesn't work at all - ERROR - module '__main__' has no attribute 'EnableNetworkingRequest' 0.0.11 is super unstable and most of the time crashes Trace32 application. When could we expect a new release version with fixed stability issues?

Wiem Wala Benayed
1 year ago

Some fixes were done to TRACE32. Please register using this link, so I can send you a recent TRACE32 version.

https://www.lauterbach.com/frames.html?register.html

JK **Jeonggyu Kim**
1 year ago
I tried this with 2 versions(0.0.6, 0.0.8) I faced "supportsANSIStyling" error on 0.0.6. I faced "module '__main__' has no attribute 'EnableNetworkingRequest'" error on 0.0.8. I want to try this with 0.0.11. but i can't find where can i download 0.0.11. Issue "EnableNetworkingRequest" is not solved yet?

JK **Jeonggyu Kim**
1 year ago
i also faced this error in 0.0.8. ERROR - Messages out of sync, connection broken. do you have solution about this error?

Wiem Wala Benayed
1 year ago

We have some issue on the build server. We are working on it. Meanwhile, I will send you a link via email for the latest version.

CS **Christoph Seitz**
8 months ago
Hello Wiem, the latest download has the following issue: ModuleNotFoundError: No module named 'typing_extensions' [PYI-212812:ERROR] Failed to execute script 't32debugadapter' due to unhandled exception! Seams that this is missing during the build.

Wiem Wala Benayed
8 months ago
Please use v 0.0.22 of t32debugadapter

G **Guo**
6 months ago
Hi, I have a question regarding using VS Code as a frontend for TRACE32. I am currently using the TRACE32 debug adapter (t32debugadapter v0.0.23) with an attach configuration. Is it possible to continuously or in real-time monitor global variable values in VS Code (similar to the Variables/Watch window behavior in TRACE32 PowerView)? For example: Can global variables be automatically refreshed while the target is running or stepping? Is real-time updating supported in the Variables or Watch view in VS Code? Are there any limitations compared to the native TRACE32 GUI? At the moment, I can see variable values, but I’m not sure whether real-time updates are fully supported or if there is a recommended setup for this. Thanks in advance!

Wiem Wala Benayed
6 months ago
No, real-time variable updates are not supported in VS Code with the TRACE32 debug adapter. Variables refresh only when you stop, step, or hit a breakpoint. No automatic updates while the target is running. This is a limitation of the Debug Adapter Protocol. For real-time monitoring, you still need TRACE32 PowerView.