Debugging Complex TS Projects in VSCode

`launch.json` Config File

This page assumes a build config based on the one in Publishing Type Definitions with a `src` and a `dist` folder.

We can use `npm link` to link `project1` to `project2` and then we can add the dist folders from both projects to `outFiles` so that they are picked up by the vscode debugger.

We also want to make sure that `sourceMaps` is set to true so that the debugger picks up the source maps in the dist folders and maps them onto the corresponding `src` folders.

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "node",
            "request": "launch",
            "name": "Launch Program",
            "program": "${workspaceFolder}/project1/dist/index.js",
            "preLaunchTask": "Build All",
            "outFiles": [
                "${workspaceFolder}/project1/dist/**/*.js",
                "${workspaceFolder}/project2/dist/**/*.js"
            ],
            "sourceMaps": true,
            "sourceMapRenames": true
        }
    ]
```

```
}
```

The launch definition can fire off a pre-launch task so we can use that to always ensure that the transpiled js code is up to date before we launch it. The below configuration defines the builds and dependencies between them

`tasks.json` Configuration

This file contains build tasks that the launch.json file can depend on before launching.

```
{
□"version": "2.0.0",
□"tasks": [
□□{
□□□"type": "typescript",
□□□"tsconfig": "projecta/tsconfig.json",
□□□"problemMatcher": [
□□□□"$tsc"
□□□],
□□□"group": "build",
□□□"label": "tsc: build - projecta/tsconfig.json",
□□□"dependsOn":["tsc: build - projectb/tsconfig.json"]
□□},
□□{
□□□"type": "typescript",
□□□"tsconfig": "projectb/tsconfig.json",
□□□"problemMatcher": [
□□□□"$tsc"
□□□],
□□□"group": "build",
□□□"label": "tsc: build - projectb/tsconfig.json"
□□},
□□{
□□□"label":"Build All",
□□□"dependsOn":[
□□□□"tsc: build - projecta/tsconfig.json",
□□□□"tsc: build - projectb/tsconfig.json"
□□□]
□□}
□]
```

```
}
```

We can define two typescript build operations - linking them together with dependencies and we can also define a `Build All` job which is just a dummy job that doesn't do anything but requires the ts builds to complete before it can be considered complete itself.

Revision #4
Created 2022-10-27 10:06:36 UTC by James
Updated 2024-01-21 14:54:02 UTC by James