CONTENTS

TIPS – Visual Studio Code & Igor Pro MCP Server 2

Igor Pro と Visual Studio Code のインストール 2

Igor Pro MCP Server のインストール 2

Igor Pro の設定 3

Visual Studio Code の設定 3

Igor Pro への接続 5

実行例 5

TIPS – Visual Studio Code & Igor Pro MCP Server

Visual Studio Code で Igor Pro MCP Server を使うまでのセットアップの手順を詳しく説明します。

ここでの環境は：
- Windows 11
- Igor Pro 10.05
- Visual Studio Code 1.147

Igor Pro と Visual Studio Code のインストール

Igor Pro は通常のインストールを行います。
Ifor Pro MCP Server と連携する必要があるため、10.05 以降をインストールしてください。
インストールは管理者ユーザーではなく、通常の Windows ユーザーで行います。

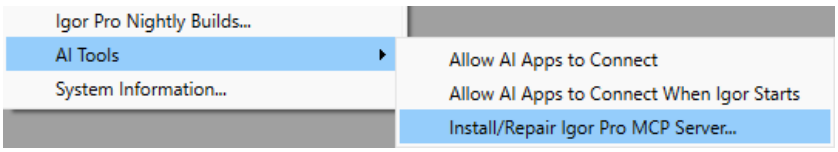
Visual Studio Code のインストールも通常のインストールで行います。

セキュリティの関係から、管理者ユーザーでは動作しません。

Igor Pro MCP Server のインストール

このサーバーは、現在の Windows ユーザー向けにインストールされます。
インストーラーによるインストールでは、管理者権限は必要ありません。

Igor Pro 10.05 以降からインストールするには、Help→AI Tools→Install/Repair Igor Pro MCP Server を選択してください。



インストーラーが起動するので、順番に設定を行いながら進みます。



サーバーインストーラーは、デフォルトで次のフォルダーにサーバーをインストールします。

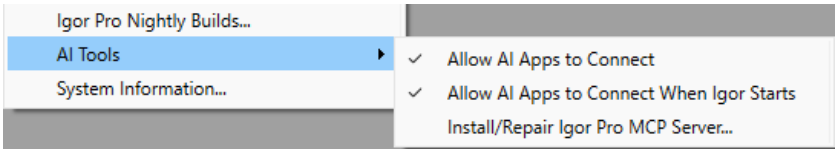
```
%LOCALAPPDATA%\Programs\WaveMetrics\Igor Pro MCP Server
```

インストーラーで別のフォルダーを選択することができます。
その場合は、Visual Studio Code の設定を行う時に、デフォルトのフォルダーの代わりにそのフォルダーを指定してください。

ここではデフォルトの場所で進めます。

Igor Pro の設定

Igor Pro を起動し、Help→AI Tools→Allow AI Apps to Connect When Igor Starts を選択します。
これで、Igor Pro を起動したときに、自動的に Visual Studio Code から接続できるようになります。



Visual Studio Code の設定

Visual Studio Code では Github Copilot 機能を使います。
最新のバージョン（2026 年 11 月時点）の Visual Studio Code には既に含まれるため、Github Copilot のインストール処理は必要ありません。
古いバージョンを使っている場合は、画面左のメニューバーから拡張機能を選択して、Github Copilot を追加してください（その場合、下記の設定方法とは手順が異なります）。

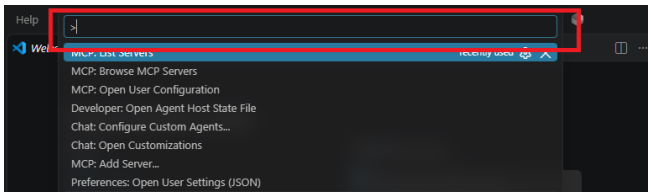
1. igor-mcp-server.exe のある場所をフルパスでコピーします。

   例えば、次のようになります。

```
C:\Users\ユーザー名\AppData\Local\Programs\WaveMetrics\Igor Pro MCP Server\igor-mcp-server.exe
```

2. Visual Studio Code で Ctrl+Shift+P キーを押して、コマンドを入力できるようします（画面上部にコマンドの入力エリアが表示されます）。

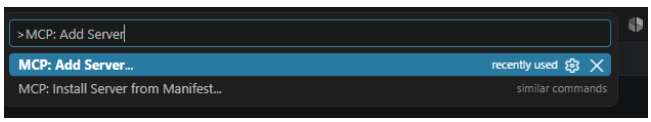
   過去に入力したことがあるコマンドがその下にリストされます。



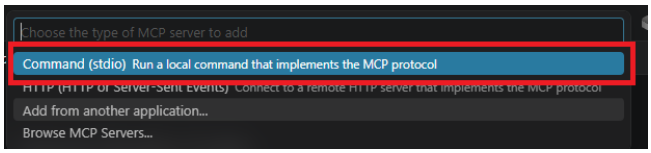
3. コマンド入力エリアに

   `MCP: Add Server`

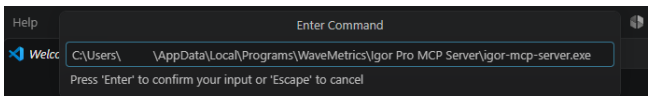
   と入力します。



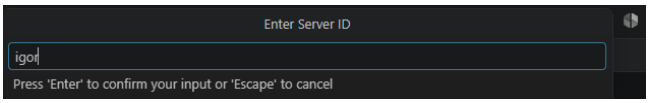
4. プルダウンメニューから Command (stdio) Run a local command that implements the MCP protocol を選択します。



5. 入力エリアにパスを貼り付け、Enter キーを押します。

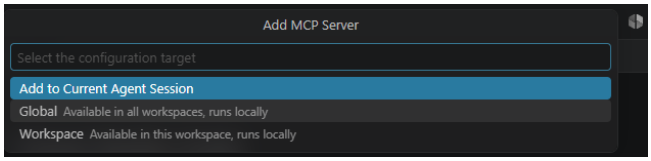


6. 次にその MCP Server の ID（名前）を付けるように促してきますので、「igor」と入力します。

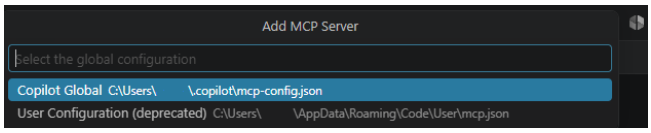


7. MCP Server をどの環境で動作させるかを聞いてきます。
このセッションのみに対して、すべてのワークスペースに対して、このワークスペースに対して、から選択できます。

   ここでは「Global」を設定した前提で話を進めます。



8. Copilot を使うので、Copilot を選択します。



9. これで設定は完了です。

   設定された mcp-config.json ファイルがエディタに表示されます。

   このファイルは

   `C:\Users\ユーザー名\.copilot`

   にあります。



10. この操作では、Visual Studio Code がパス内の空白文字をうまく処理できず、空白文字以降を “args” として処理してしまっています。

    次のように修正します。

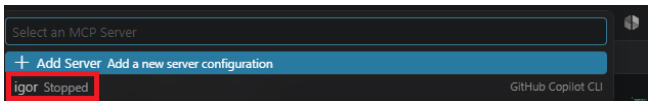
```
    "type": "stdio",
    "command": "C:\\Users\\ユーザー名\\AppData\\Local\\Programs\\WaveMetrics\\Igor Pro MCP Server\\igor-mcp-server.exe"
```

11. 今後、MCP Server が正しく使われるかを確認するため、Visual Studio Code を再起動します。

    Ctrl+Shift+P キーを押して、コマンド入力エリアに

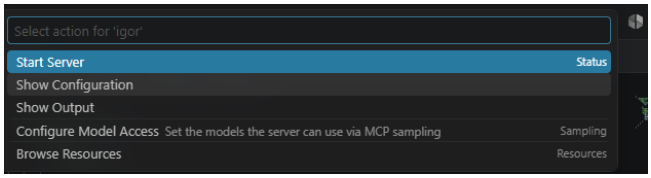
    `MCP: List Servers`

    と入力すると「igor」というエントリが追加されています。



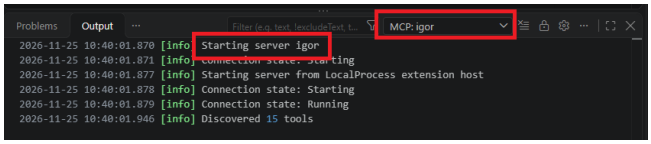
12. 前のステップの「igor」というエントリをクリックすると、ポップアップメニューが表示されます。

    ここで Start Server を選択します。



13. 画面下部の出力に起動状態が表示されます。

    起動に失敗している場合は、何らかのエラーが表示されているはずです。



Igor Pro への接続

サーバーは、すでに実行中の Igor Pro に接続します。

Igor Pro が起動されていないと、「Igor Pro が見つからない」というアラートが表示されます。

実行例

Visual Studio Code 内の Copilot Chat のプロンプト（画面右下）で、例えば、次のように入力します。

「Igor Pro 内でノイズを含む 2 次元サンプルデータを作成し、それを 2 次元 FFT して、結果を視覚的に表示して」



実行が始まると、操作の許可を求めるメッセージがされることがあります。
これらを許可していくと、次々と操作が実行されていきます。

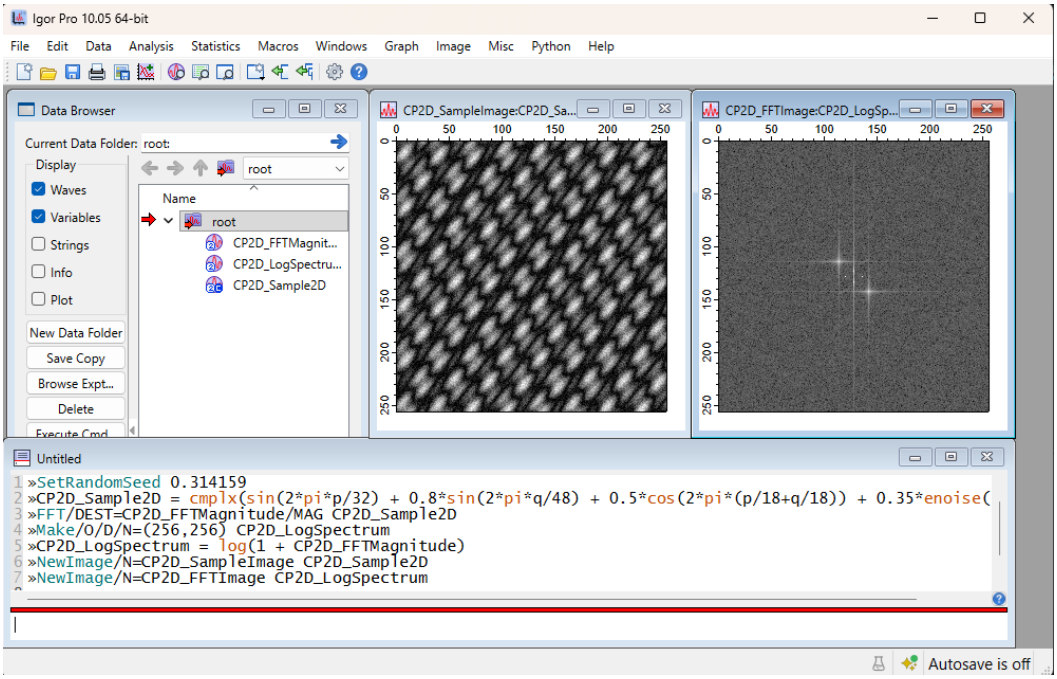
Igor Pro の画面を見ると、ウェーブが生成され、各種処理や表示が行われていることがわかります。

すべてが終了すると、Copilot の Chat 画面では、例えば、次のような出力が表示されます。

Igor Pro 内で作成・表示しました。現在、FFT 画像のウィンドウが前面にあります。

- CP2D_Sample2D：256×256 の 2 次元サンプル。複数の正弦波に一様ノイズを加えています。
- CP2D_FFTMagnitude：サンプルの 2 次元 FFT 振幅
- CP2D_LogSpectrum：見やすさのため、振幅を log(1 + 振幅) に変換した表示用データ
- 表示ウィンドウ：CP2D_SampleImage、CP2D_FFTImage

Igor Pro 側を見ると、次のように、AI が実行した結果が表示されます。


Igor Pro 10.05 64-bit
File Edit Data Analysis Statistics Macros Windows Graph Image Misc Python Help
Data Browser
Current Data Folder: root:
Display
Waves
Variables
Strings
Info
Plot
New Data Folder
Save Copy
Browse Expt...
Delete
Name
root
CP2D_FFTMagnit...
CP2D_LogSpectru...
CP2D_Sample2D
CP2D_SampleImage:CP2D_Sa...
CP2D_FFTImage:CP2D_LogSp...
Untitled
»SetRandomSeed 0.314159
»CP2D_Sample2D = cmplx(sin(2*pi*p/32) + 0.8*sin(2*pi*q/48) + 0.5*cos(2*pi*(p/18+q/18)) + 0.35*enoise(
»FFT/DEST=CP2D_FFTMagnitude/MAG CP2D_Sample2D
»Make/O/D/N=(256,256) CP2D_LogSpectrum
»CP2D_LogSpectrum = log(1 + CP2D_FFTMagnitude)
»NewImage/N=CP2D_SampleImage CP2D_Sample2D
»NewImage/N=CP2D_FFTImage CP2D_LogSpectrum
Autosave is off