CONTENTS

ビジュアルヘルプ – Igor Pro MCP Server 2

Igor Pro における AI の機能 2

始める前に 2

システム要件 3

サーバーのインストール 3

AI アプリの設定 4

Claude Code 5

Claude Desktop 5

Cursor 6

Visual Studio Code 6

その他の AI アプリ 7

Igor Pro への接続 7

サーバーのアップデート 8

サーバーのアンインストール 9

トラブルシューティング 9

IT 管理者向け 10

インストールと署名 10

サーバーの実行を許可する 11

ポリシー設定 13

サーバーコマンドラインオプション 14

ビジュアルヘルプ – Igor Pro MCP Server

Igor Pro MCP Server は、AI チャットアプリをコンピューター上の Igor Pro に接続します。
利用可能なアプリには、Claude Code や Desktop、Cursor、Visual Studio Code 内の GitHub Copilot などがあります。
AI は、Igor Pro コマンドの実行、関数の呼び出し、プロシージャファイルのコンパイルを行うことができます。
また、グラフの確認や Igor Pro のドキュメントの検索も可能です。

MCP とは「Model Context Protocol」の略で、AI アプリがツールに接続するために使うオープンスタンダードです。
AI アプリがサーバーを起動し、そのサーバーと通信します。
サーバーは、さらに Igor Pro と通信します。

Igor Pro MCP Server は Windows でのみ動作します。

Igor Pro における AI の機能

サーバーを介して、AI エージェントは以下のことが可能です。

- Igor Pro のコマンドや操作を実行し、組み込み関数やユーザー定義関数を呼び出す
- プロシージャをコンパイルし、コンパイルエラーを報告するとともに、プロシージャファイルを開いたり閉じたりする
- Igor Pro のウィンドウを一覧表示し、グラフ、テーブル、レイアウト、パネル、または Gizmo ウィンドウの画像をキャプチャする
- 現在のエクスペリメントやシステム情報を報告し、処理中か、それともダイアログを待機中かを報告する
- 自身が開始したコマンドや関数の呼び出しが長時間かかっている場合、それを停止する
- 組み込みのコマンドや関数については、ドキュメントを参照する
- Igor Pro を起動、終了する

AI エージェントは、コマンドを実行したり関数を呼び出したりする時、その結果やエラーメッセージを直接受け取ります。
これらのエラーメッセージは、Igor Pro に表示されるものよりも詳細な場合が多く、通常、履歴エリアには表示されません。

始める前に

以下の点に留意してください。

- **AI エージェントは、ユーザーの許可を得て動作します。**
  Igor Pro のコマンドを使うと、データの作成・変更・削除、ファイルの書き込み・削除、プロシージャに含まれる任意のコードの実行が可能です。
  Igor Pro でコマンドを実行できる AI エージェントは、これらすべての操作を行うことができます。
  特に、まだ保存していない作業が含まれているエクスペリメントでは、AI エージェントが提案する内容をよく確認し、ツール呼び出しを承認する時は AI アプリの設定に従ってください。

- **AI アプリは、その AI サービスに情報を送信します。**
  サーバーを経由するコマンド、結果、プロシージャコード、ウィンドウの画像は、AI エージェントとの会話の一部となります。
  AI アプリは、その AI プロバイダーの利用規約に従って、チャットをそのプロバイダーに送信します。
  サーバー自体は、WaveMetrics に何も送信しません。
  インターネットへの唯一の接続は、docs.wavemetrics.com でドキュメントを検索するためだけです。

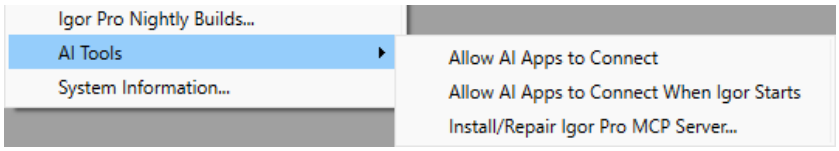
- **AI アプリが接続できるタイミングは、ユーザーが決定します。**
  「Igor への接続」セクションで説明されているように、Igor Pro はユーザーが許可した場合、またはその Igor Pro インスタンスを起動した AI エージェントが許可した場合にのみ、接続を受け入れます。

- **Igor Pro に接続できるのはあなただけです。**
  Igor Pro は、同じ Windows ユーザーとして実行されているプログラムからの接続のみを受け付けます。
  管理者権限で実行されている場合（例えば、「Run as administrator」で起動した場合など）は、接続を受け付けません。
  管理者アカウントで Windows にサインインしても、接続が妨げられることはありません。

システム要件

- Windows 10 以降

- Igor Pro 10.05 以降
  それ以前のバージョンでは、サーバーからの接続を受け付けません。
  MCP サーバーをサポートする Igor Pro のリリースには、Help→AI Tool メニューがあります。
  この展開可能なメニューを使うと、サーバーのインストールや修復、および接続の管理を行うことができます。

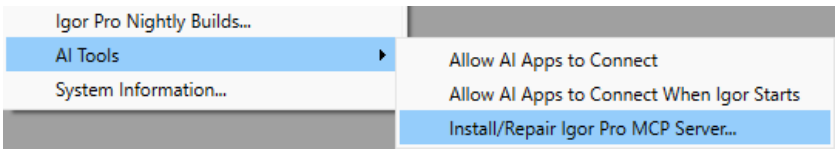


- Claude Code や Desktop、Cursor、Visual Studio Code の GitHub Copilot など、MCP サーバーに対応した AI アプリ

サーバーのインストール

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。



Igor Pro のメインウィンドウ下部のステータスバーにある「AI 接続」ボタン（ ）をクリックすると、同じメニューが表示されます。
また、AI アプリの接続を許可しようとした時にサーバーがインストールされていない場合、サーバーのインストールを提案します。

このメニューから初めてサーバーをインストールした場合、Igor Pro に戻った時に、AI アプリの接続を許可するかどうかを尋ねられます。
すでに許可されている場合は、この確認は行われません。



Igor Pro 10.05 以降のインストーラーでも、サーバーをインストールできます。
Igor Pro インストーラーの最終ページで、Install the Igor Pro MCP Server for me にチェックを入れてください。

また、サーバーインストーラー「setupIgorProMCPServer.exe」をダウンロードして、自身で実行することもできます。

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、AI アプリの設定を行う時に、デフォルトのフォルダーの代わりにそのフォルダーを使ってください。

サーバーのインストールが完了したら、次のセクションの説明に従って AI アプリを設定してください。
AI アプリがすでに実行中の場合は、再起動してください。

AI アプリの設定

各 AI アプリには独自の設定ファイルがあり、そこで MCP サーバーの起動方法を指定します。
以下の例では、サーバー名を「igor」としています。
インストールされた 1 つのサーバーで、Igor Pro 10 以降のすべてのバージョンに対応できるため、この名前を推奨します。

これらの例では、LOCALAPPDATA 環境変数を通じてサーバーのデフォルトの場所を参照しているため、この設定はすべての Windows ユーザーで共通して機能します。
サーバーを別のフォルダーにインストールした場合は、そのフォルダー内の igor-mcp-server.exe へのフルパスに置き換えてください。
例えば、サーバーを `D:\Tools\Igor Pro MCP Server` にインストールした場合、Claude Code コマンドは次のようになります。

```
claude mcp add --scope user igor -- "D:\Tools\Igor Pro MCP Server\igor-mcp-server.exe"
```

JSON 設定ファイルでは、各バックスラッシュを 2 回記述してください。

```
"command": "D:\\Tools\\Igor Pro MCP Server\\igor-mcp-server.exe"
```

JSON 設定ファイルにすでに他のサーバーが記載されている場合は、それらの隣に “igor” のエントリを追加し、エントリ同士をコンマで区切ってください。
最後のエントリの後にコンマを付けないでください。
例えば：

```
{
  "mcpServers": {
    "other-server": {
      "command": "C:\\Tools\\other-server.exe"
    },
    "igor": {
```

```
      "command": "D:\\Tools\\Igor Pro MCP Server\\igor-mcp-server.exe"
    }
  }
}
```

設定を変更したら、Igor Pro で変更内容を保存し、AI アプリを再起動してください。
Visual Studio Code では、代わりにサーバーのみを再起動することも可能です。
詳細は「Visual Studio Code」のセクションを参照してください。

AI アプリの設定は頻繁に変更されます。
以下の手順が実際の画面と異なる場合は、アプリのセクションの最後にあるリンクから、そのアプリ専用の MCP サーバー追加手順を参照してください。

Claude Code

PowerShell で次のコマンドを実行します。

```
claude mcp add --scope user igor -- "$env:LOCALAPPDATA\Programs\WaveMetrics\Igor Pro MCP Server\igor-mcp-server.exe"
```

Claude Code がパスを保存する前に、PowerShell が `$env:LOCALAPPDATA` をユーザー自身のフォルダーに置き換えます。
--scope user を指定すると、すべてのプロジェクトでサーバーが利用可能になります。

サーバーを確認するには、`claude mcp list` を実行してください。
igor が接続中と表示されるはずです。

詳細は、Claude Code のドキュメントにある Connect Claude Code to tools via MCP を参照してください。

Claude Desktop

Claude Desktop ではフルパスが必要ですので、以下の USER_NAME を自分の Windows ユーザー名に置き換えてください。
Claude Desktop で、File→Settings の順に選択します。
Desktop アプリの下にある Extensions を選択し、Manage をクリックしてから、Local MCP Servers の下にある Edit Config をクリックします。
Claude Desktop は、ファイルエクスプローラーに claude_desktop_config.json ファイルを表示します。
そのファイルをテキストエディタで開き、以下を追加してください。

```
{
  "mcpServers": {
    "igor": {
      "command": "C:\\Users\\USER_NAME\\AppData\\Local\\Programs\\WaveMetrics\\Igor Pro MCP Server\\igor-mcp-server.exe"
    }
  }
}
```

ファイルにプレファレンスなどの他の設定がすでに含まれている場合は、mcpServers をそれらの隣にファイルの最上位に配置し、各エントリをコンマで区切ってください。
ファイルにすでに mcpServers が含まれている場合は、そこに igor エントリを追加し、他のエントリとコンマで区切ってください。

Claude Desktop を、通知領域のアイコンも含めて完全に終了させ、再度起動してください（フロントエンドのアプリだけでなく、タスクマネージャーでバックグラウンドプロセスも停止して、再度起動してください。）。

詳細は、Claude ヘルプセンターの Getting Started with Local MCP Servers on Claude Desktop を参照してください。

Cursor

`%USERPROFILE%\.cursor\mcp.json` に以下を追加してください。

```
{
  "mcpServers": {
    "igor": {
      "command": "${env:LOCALAPPDATA}\\Programs\\WaveMetrics\\Igor Pro MCP Server\\igor-mcp-server.exe",
      "args": ["--"]
    }
  }
}
```

`"args": ["--"]` という行は残しておいてください。
サーバーに引数が指定されていない場合、Cursor はスペースごとにコマンドを分割しますが、これにより Igor Pro MCP Server のパスが途切れてしまいます。
`--` という引数は、サーバーには何の影響も与えません。

ファイルにすでに他のサーバーが記載されている場合は、その mcpServers オブジェクトに igor というエントリを追加し、他のエントリとカンマで区切ってください。
サーバーを確認するには、Customize を選択し、次に MCPs を選択してください。

詳細は、Cursor のドキュメントにある Model Context Protocol (MCP) を参照してください。

Visual Studio Code

Visual Studio Code の GitHub Copilot は、`%USERPROFILE%\.copilot\mcp-config.json` から MCP サーバーを読み込みます。
そのファイルに以下を追加してください。

```
{
  "mcpServers": {
    "igor": {
      "command": "${LOCALAPPDATA}\\Programs\\WaveMetrics\\Igor Pro MCP Server\\igor-mcp-server.exe"
    }
  }
}
```

ファイルにすでに他のサーバーが記載されている場合は、その mcpServers オブジェクトに igor のエントリを追加し、他のエントリとカンマで区切ってください。
また、コマンドパレット（Ctrl+Shift+P）から MCP: Add Server を実行し、Global を選択してから Copilot Global を選択することもできます。

Visual Studio Code のチャット内の Local エージェントは、代わりにユーザープロファイル内の古い mcp.json ファイルを読み取ります。
そのファイルは、異なる形式と異なる変数構文を使っています。

```
{
  "servers": {
    "igor": {
      "type": "stdio",
      "command": "${env:LOCALAPPDATA}\\Programs\\WaveMetrics\\Igor Pro MCP Server\\igor-mcp-server.exe"
    }
  }
}
```

mcp.json にすでに他のサーバーが記載されている場合は、その servers オブジェクトに igor のエントリを追加し、他のエントリとカンマで区切ってください。

Visual Studio Code を再起動せずにサーバーを再起動するには、コマンドパレット（Ctrl+Shift+P）から MCP: List Servers を実行し、igor を選択してから、Stop Server を選択し、続いて Start Server を選択します。

サーバーが停止する前に作業内容を保存してください。
Visual Studio Code の GitHub Copilot が Igor MCP サーバーを停止すると、AI が起動した Igor もすべて、保存の確認を求めずに閉じられます。
これは、サーバーを停止または再起動したとき、Visual Studio Code を終了したとき、例えば mcp.json ファイルの編集など、MCP サーバーの設定が変更された後に発生します。
このような変更後、AI が応答を完了したとき、または次のメッセージを送信したときに、たとえユーザー自身がサーバーを停止していなくても、Visual Studio Code はチャット用のすべての MCP サーバーを再起動します。
Igor Pro は、ユーザー自身が起動した場合は開いたままになります。
他の AI 環境ではこの挙動は確認されていませんが、MCP サーバーを停止する前に、常に Igor Pro 内の作業を保存しておくことを推奨します。

詳細は、Visual Studio Code のドキュメントにある Add and use MCP servers in VS Code を参照してください。

その他の AI アプリ

ローカル（stdio）MCP サーバーをサポートしている他の AI アプリも、このサーバーを利用できます。
アプリに、igor-mcp-server.exe へのフルパスをコマンドとして指定してください。
サーバーには引数を指定する必要はありません。
アプリのログに `...\WaveMetrics\Igor` を実行しようとしたと表示されている場合、アプリがスペースでパスを分割しています。
Cursor の場合と同様に、引数として `--` を追加してください。

Igor Pro への接続

サーバーは、すでに実行中の Igor Pro に接続するか、AI が代わりに Igor Pro を起動します。

AI が Igor Pro を起動すると、Igor Pro はサーバーからの接続を受け入れる準備が整います。
AI は、別のバージョンを起動するよう指示しない限り、最も新しくインストールされたバージョンの Igor Pro を起動します。

自分で起動した Igor Pro を使うには、AI アプリがそれに接続できるようにする必要があります。
Igor Pro は、ユーザーが許可しない限り接続を受け付けません。
Help→AI Tools から、またはステータスバーの AI connections ボタンをクリックして表示されるメニューから、以下の項目のいずれかを選択してください。

- 「Allow AI Apps to Connect」を有効にすると、この Igor Pro が終了するまで接続が可能になります。
  再度選択すると許可が解除され、接続中の AI アプリをすべて切断します。

- 「Allow AI Apps to Connect When Igor Starts を選択すると、現在や今後 Igor Pro を起動するたびに接続が許可されます。
  再度このオプションを選択すると、今後起動する Igor Pro セッションでは接続が許可されなくなります。
  実行中の Igor Pro では、「Allow AI Apps to Connect」を再度選択するか、Igor Pro を終了するまで接続が引き続き許可されます。

AI connections ボタンは、AI アプリがこの Igor Pro に接続できるかどうかを示します。
このボタンの表示は、AI アプリが接続できない場合は 、接続できる場合は  となります。
詳細を確認するには、ボタンにマウスを合わせてください。
ボタンを非表示または表示するには、ステータスバーを右クリックし、Hide AI Apps Connection または Show AI Apps Connection を選択してください。

接続を許可している Igor Pro が複数ある場合、サーバーは最も最近起動された Igor Pro に接続します。
その後、たとえ別の Igor Pro を起動しても、その Igor Pro が実行中で接続を許可している限り、サーバーはその Igor Pro に接続し続けます。
AI に特定の Igor Pro を使用させるには、その Igor Pro でのみ接続を許可してください。

AI が使っていた Igor Pro が終了したり、接続を許可しなくなったりした場合、別の Igor Pro が接続を許可していても、サーバーは黙ってその Igor Pro に切り替えることはありません。
サーバーは AI に対し、何も送信しなかったことや、別の Igor Pro が独自のエクスペリメントを持っていることを通知します。
その前に最初の Igor Pro への接続を再び許可しない限り、AI の次のリクエストは別の Igor Pro に送信されます。

サーバーのアップデート

「サーバーのインストール」のセクションの手順に従ってサーバーをインストールした場合、Igor がサーバーを最新の状態に保ちます。
IT 部門がサーバーを管理している場合は、IT 部門が更新処理を行います。
「IT 管理者向け」のセクションを参照してください。
Igor Pro の更新を確認する時（自動の場合、または Help→Updates for Igor Pro を選択した場合）、同時に新しい Igor Pro MCP Server の有無も確認します。
新しいサーバーが利用可能な場合、Igor Pro はそのインストールを提案します。
新機能を確認するには、そのダイアログで Show release notes をクリックするか、WaveMetrics ウェブサイトの Igor Pro MCP Server ページを参照してください。

Igor Pro の新しいバージョンには、その都度サーバーインストーラーが含まれています。
アップデートサーバーと使用中の Igor Pro の両方に新しいサーバーがある場合、２つのうち新しい方を提示します。
Igor Pro は、アップデートサーバーに接続できない場合でも、製品に同梱されているサーバーを提示します。
Miscellaneous Settings ダイアログで自動アップデートチェックを無効にしている場合でも、起動時に、バージョンごとに１回ずつ、製品に同梱されている新しいサーバーを提示します。

AI が Igor Pro を起動した時、サーバーの更新を自動的に確認しません。
これは、ユーザーがその場にいて応答できない可能性があるためです。
確認するには、Help→Updates for Igor Pro を選択してください。

サーバーの更新後、次のように表示されます。
「Igor Pro MCP サーバーが更新されました。作業内容を保存してから、AI チャットアプリを再起動して、新しいバージョンが使われるようにしてください。AI アプリを再起動すると、AI によって起動された Igor Pro が閉じられる場合があります。」
AI アプリを再起動するまでは、AI アプリは引き続き古いバージョンを使用し続けます。

サーバーのアンインストール

サーバーをアンインストールするには、Windows の設定を開き、「アプリ」に移動して、「Igor Pro MCP Server」を見つけ、「アンインストール」を選択します。
その後、AI アプリの設定から「igor」のエントリを削除してください。

Igor Pro をアンインストールしても、サーバーはアンインストールされません。

トラブルシューティング

AI アプリで「サーバーの起動に失敗しました」と表示される。

AI アプリの設定にあるパスが、インストール先のフォルダー内にある「igor-mcp-server.exe」を指しているか確認してください。
サーバーを別のフォルダーにインストールした場合、サンプルに記載されているデフォルトのパスは適用されません。
パスが正しい場合でも、所属組織によってユーザーフォルダー内のプログラムがブロックされている可能性があります。
「IT 管理者向け」のセクションを参照してください。

AI アプリで「Igor が実行されていないか、Igor に接続できない」と表示される。

以下の原因を確認してください：

Igor Pro が古すぎます。
Igor Pro はバージョン 10.05 以降である必要があります。

Igor Pro が接続を許可していません。
Help→AI Tools→Allow AI Apps to Connect を選択してください。
「Igor Pro への接続」のセクションを参照してください。
サーバーは、接続を許可していない Igor Pro を認識できません。
Igor Pro が見つからない場合、サーバーは 2 つ目の Igor Pro を起動する代わりに、AI に対してユーザーに接続の許可を求めるよう指示します。

Igor Pro が管理者として実行されています。
管理者権限で実行されている場合（例えば、「管理者として実行」で起動した場合など）、接続を受け付けません。
Windows の管理者アカウントを使っても問題ありません。
管理者権限で実行されている Igor Pro では、接続を許可しようとした時、または起動時の履歴エリアで、「接続を受け付けられません」というメッセージが表示されます。

貴組織では AI ツールが無効化されています。
Help→AI Tools に「Your IT department has disabled Igor AI Tools」と表示される場合、貴組織では AI アプリが Igor Pro に接続することを許可していません。
「IT 管理者向け」のセクションを参照してください。

AI アプリで、それまで使っていた「Igor」にはもはや接続できず、別の「Igor」が稼働していると表示される。

AI が使っていた「Igor」は終了したか、接続を許可しなくなったため、別の「Igor」が接続を許可しています。
どちらの Igor Pro にも何も送信されていません。
もう一方の Igor Pro には独自のエクスペリメントが設定されているため、AI のデータ、ウィンドウ、プロシージャはそこに存在しません。
最初の Igor Pro で作業を続行するには、その Igor Pro で再び接続を許可してください。
そうしないと、AI の次のリクエストは別の Igor Pro に送信されてしまいます。
「Igor Pro への接続」のセクションを参照してください。

AI からのリクエストに時間がかかる。

Igor Pro は AI のコマンドを 1 つずつ実行します。
一部のコマンドは、ユーザーの応答を待ちます。
モーダルダイアログが表示されたり、デバッガーが開いたり、PauseForUser が呼び出されたりする場合があります。
Igor Pro で応答するか、AI にリクエストを停止するよう指示してください。
また、Ctrl+Break キーを押すことで、実行中のコマンドを停止することもできます。

AI が、サーバーに対して行った変更を認識しない。

Igor Pro で作業内容を保存し、サーバーのインストールまたは更新が完了した後、AI アプリを再起動してください。

WaveMetrics のサポートからログの提出を求められた。

サーバーはログファイルを作成しません。
診断メッセージを取得するには、AI アプリの設定でサーバーの引数に「--verbose」を追加し（「サーバーのコマンドラインオプション」のセクションを参照）、作業内容を保存してから、AI アプリまたはサーバーを再起動し、問題が発生した操作を繰り返してください。
サーバーはメッセージを標準エラー出力に出力し、AI アプリがそれらを記録します。
Visual Studio Code では、コマンドパレット（Ctrl+Shift+P）から「MCP: List Servers」を実行し、サーバーを選択して、その出力を確認してください。
Claude Desktop では、これらのメッセージは `%LOCALAPPDATA%\Claude\logs\mcp-server-igor.log` に書き込まれます。
メッセージには、AI が Igor Pro に送信したコマンドや関数呼び出し、AI が開いたファイルのパスが含まれていますので、送信する前に内容を確認してください。
作業が完了したら、--verbose オプションを削除してください。

IT 管理者向け

インストールと署名

サーバーインストーラー「setupIgorProMCPServer.exe」は、ユーザーごとにインストールされます。
管理者権限は必要なく、HKEY_CURRENT_USER 配下にインストール情報が登録されます。
デフォルトのフォルダーは次のとおりです。

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

サーバーの実行ファイルは「igor-mcp-server.exe」です。
サーバーとそのインストーラーは、Igor Pro 10 に署名しているのと同じ発行元である WaveMetrics（CN=WaveMetrics、O=WaveMetrics、L=Portland、S=Oregon、C=US）によって署名されています。

サーバーをユーザーに展開するには、各ユーザーのコンテキストでインストーラーを実行します。
例えば、Microsoft Intune や Configuration Manager を使って、そのユーザーとしてインストールするように設

定します。
インストーラーは、以下のコマンドラインオプションをサポートしています。

オプション	効果
`/q`	インターフェイスを表示せずに、静かにインストールします。
`/install:folder`	デフォルトのフォルダーではなく、別のフォルダーにインストールします。フォルダー名にスペースが含まれる場合は、オプション全体を引用符で囲んでください（例：`"/install:D:\AI Tools\Igor Pro MCP Server"`）。
`/log:file`	インストーラーのログを、ファイル名を含むフルパスでファイルに保存します。

例えば、コマンドプロンプトから次のように入力します：

```
start /wait setupIgorProMCPServer.exe /q
echo %ERRORLEVEL%
```

このインストーラーが正常に完了した場合、終了コードは 0 になります。
終了コード 6 は、インストーラーが失敗し、変更が一切行われなかったことを意味します。
その原因としては、次のようなものが考えられます。

そのユーザーにはすでに新しいバージョンのサーバーがインストールされている

ServerPath ポリシーが設定されている（以下の「ポリシー設定」のセクションを参照）。

このインストーラーは InstallMate を使って構築されており、その他の終了コードは InstallMate のものです。
InstallMate のドキュメントにある Exit codes の項を参照してください。

サーバーを静かにアンインストールするには、以下の QuietUninstallString 値にあるコマンドを実行してください。

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Uninstall\{75C12DF8-4C8C-4E9B-90FD-594D3531B07D}
```

コマンドは次のとおりです。

```
"%LOCALAPPDATA%\Installer\{75C12DF8-4C8C-4E9B-90FD-594D3531B07D}\Setup.exe" /remove /q
```

サーバーの実行を許可する

組織で AppLocker または App Control for Business を使ってユーザーフォルダー内のプログラムをブロックしている場合は、以下の手順に従ってサーバーを許可してください。

AppLocker

以下のルールのいずれかを適用して、サーバーへのアクセスを許可してください。

WaveMetrics のパブリッシャールール。
このルールを設定すると、サーバーはどのフォルダーからでも実行できるようになります。
パブリッシャーレベルでルールを作成し、製品名、ファイル名、バージョンには「*」を指定します。
Igor64.exe からルールを作成する場合は、スライダーを Publisher の位置に移動してください。
製品名レベルでは、このルールは Igor64.exe の製品名である「IGOR Pro」を使いますが、これはサーバーの製品名である「IGOR Pro MCP SERVER」とは一致しません。

デフォルトのフォルダーに対するパスルール。
AppLocker のパスルールでは `%LOCALAPPDATA%` は使えないため、次のパスを使ってください。

```
%OSDRIVE%\Users\*\AppData\Local\Programs\WaveMetrics\Igor Pro MCP Server\*
```

例えば、この AppLocker のパブリッシャールールでは、すべてのユーザーに対して、サーバーと Igor Pro 10 の両方が許可されています。

```
<FilePublisherRule Id="5f0ad0b4-0b8b-4a1e-9a4a-1c9f8d2e6b31" Name="WaveMetrics" Description=""
UserOrGroupSid="S-1-1-0" Action="Allow">
  <Conditions>
    <FilePublisherCondition PublisherName="O=WAVEMETRICS, L=PORTLAND, S=OREGON, C=US"
ProductName="*" BinaryName="*">
      <BinaryVersionRange LowSection="*" HighSection="*" />
    </FilePublisherCondition>
  </Conditions>
</FilePublisherRule>
```

Igor Pro 9 以前のバージョンは、別のパブリッシャー名で署名されていたため、このルールではそれらを許可しません。

サーバーのアンインストーラーには署名がないため、パブリッシャールールによって許可されません。
ユーザーが Windows の設定からサーバーをアンインストールできるようにするには、次の AppLocker パスルールでも許可するように設定してください。

```
%OSDRIVE%\Users\*\AppData\Local\Installer\{75C12DF8-4C8C-4E9B-90FD-
594D3531B07D}\Setup.exe
```

また、アンインストーラーは、InstallMate のクリーンアップヘルパーである TinDel.exe を `%TEMP%` にコピーして実行します。
このヘルパーには Tarma Software Research Ltd による署名が付いているため、WaveMetrics のパブリッシャールールによって許可されません。
このヘルパーがブロックされた場合でも、サーバーはアンインストールされますが、インストーラーは `%LOCALAPPDATA%\Installer\{75C12DF8-4C8C-4E9B-90FD-594D3531B07D}` に残ります。
このヘルパーを許可するには、次の AppLocker パブリッシャールールを追加してください。

```
<FilePublisherRule Id="3d9f2a61-7c1e-4b8a-9f40-2e6d5b8c1a07" Name="InstallMate cleanup helper"
Description="" UserOrGroupSid="S-1-1-0" Action="Allow">
  <Conditions>
    <FilePublisherCondition PublisherName="O=TARMA SOFTWARE RESEARCH LTD, S=HERTFORDSHIRE, C=GB"
ProductName="INSTALLMATE" BinaryName="TINDEL.EXE">
      <BinaryVersionRange LowSection="*" HighSection="*" />
    </FilePublisherCondition>
  </Conditions>
</FilePublisherRule>
```

App Control for Business

パブリッシャーレベルでルールを作成します。
Igor Pro 10 の Igor64.exe から作成できます。
両方のファイルのパブリッシャーと Microsoft の認証局が同一であるため、このルールではサーバーも許可されます。

サーバーのアンインストーラーには署名がないため、パブリッシャールールでは許可されません。
また、App Control は、ユーザーが書き込み可能なフォルダーに対するパスルールを無視するため、ポリシーに Disabled:Runtime FilePath Rule Protection オプションが設定されていない限り、アンインストーラーに対するパスルールは機能しません。
ユーザーがサーバーをアンインストールできるようにするには、Intune Management Extension などの管理対象インストーラーを介してサーバーインストーラーを展開し、ポリシーで Enabled:Managed Installer オプションを設定してください。

これにより、App Control は、アンインストーラーを含め、インストーラーが書き込むファイルを許可するようになります。
ハッシュルールを使ってアンインストーラーを許可することもできますが、そのハッシュ値はサーバーのバージョンごとに変化します。

アンインストーラのクリーンアップヘルパーである TinDel.exe には、独自のルールが必要です。
これは、アンインストーラが実行時にこのファイルを `%TEMP%` に書き込むためです。
TinDel.exe に基づいてパブリッシャールールを作成してください。
サーバーをアンインストールした後も、このヘルパーは `%TEMP%` に残っているため、そのコピーからルールを作成することができます。

ポリシー設定

Igor Pro とサーバーのインストーラーは、以下の場所にあるこれらのレジストリ値を読み取ります。

`HKEY_LOCAL_MACHINE\SOFTWARE\Policies\WaveMetrics\Igor Pro MCP Server`

値	形式	効果
`ServerPath`	REG_SZ または REG_EXPAND_SZ	IT 部門がこのパスにあるサーバーを管理しています。ユーザーごとのインストーラーはインストールを拒否します。Igor Pro は Help→AI Tools→Install/Repair Igor Pro MCP Server を非表示にし、サーバーの更新を確認しません。
`DisableUserUpdates`	REG_DWORD を 1 に設定	IT 部門がユーザーごとのサーバーのインストールと更新を行います。Igor Pro はサーバーの更新を確認しません。Help→AI Tools→Install/Repair Igor Pro MCP Server は引き続き利用可能です。

両方の値が設定されている場合、`ServerPath` が優先されます。
HKEY_CURRENT_USER にある同じ値は無視されます。
いずれの値も、すでにインストールされているサーバーをアンインストールすることはありません。

Igor Pro がサーバーや他のプログラムからの接続を一切受け付けないようにするには、次の値を設定してください。

`HKEY_LOCAL_MACHINE\SOFTWARE\Policies\WaveMetrics\Igor Pro`

値	形式	効果
`JsonRpcAllowed`	REG_DWORD を 0 に設定	Igor Pro はサーバーからの接続を受け付けません。 Help→AI Tools サブメニューには、無効化された項目 Your IT department has disabled Igor AI Tools のみが表示され、ステータスバーの AI connections ボタンを非表示にします。 ステータスバーの右クリックメニューには、Show AI Apps Connection という項目がありません。 Igor Pro が起動時に接続を許可するように設定されている場合でも、履歴エリアには接続がブロックされていると表示されます。

サーバーコマンドラインオプション

AI アプリがサーバーを起動するため、通常はオプションを指定する必要はありません。
以下のオプションは、特別な場合に利用可能です。

オプション	**効果**
`--igor-version V`	バージョン番号が「V」で始まる Igor Pro（例：10 や 10.05 など）を使うことを推奨します。
`--verbose`	診断メッセージを標準エラー出力に書き出します。メッセージが表示される場所は AI アプリによって異なります。例えば、Claude Desktop の場合は、`%LOCALAPPDATA%\Claude\logs\mcp-server-Igor.log` に書き出されます。
`--server-version`	サーバーのバージョンを表示して終了します。

オプションを使うには、AI アプリの設定で、サーバーの args リストにそのオプションを追加してください。
リスト内の項目はコンマで区切ってください。
サーバーのエントリに args 行がない場合は、コマンドラインの後に args 行を追加し、コマンドラインの末尾にコンマを付けてください。
例えば：

```
"igor": {
  "command": "C:\\Users\\USER_NAME\\AppData\\Local\\Programs\\WaveMetrics\\Igor Pro MCP
Server\\igor-mcp-server.exe",
  "args": ["--igor-version", "10"]
}
```

Cursor では、"args": ["--igor-version", "10", "--"] のように -- の前にオプションを記述してください。
サーバーは、-- の後に続くオプションを無視します。