

IDL 9.0 新機能紹介

NV5 Geospatial 株式会社

N|V|5

IDL 9.0の新機能

- ログおよびインストールパスの変更
- 新機能
- 機能向上
- ライブラリの更新

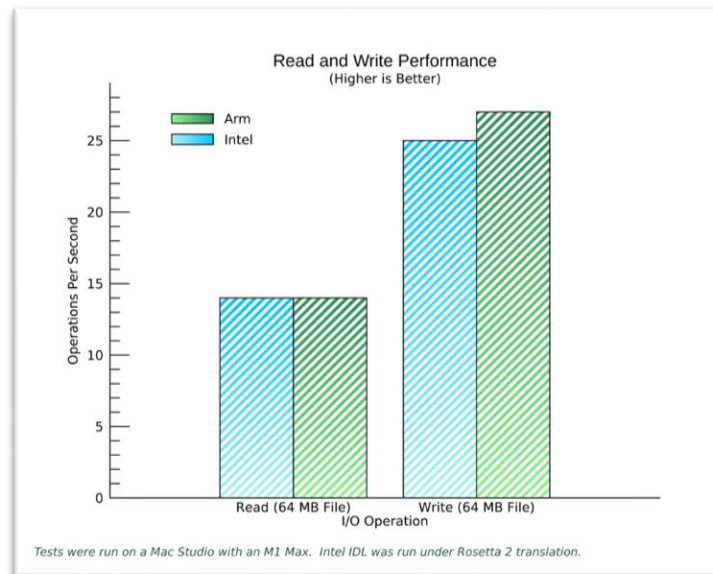
ロゴおよびインストールパスの変更

- IDL9.0からアイコンとスプラッシュ画面が変更されました。
- デフォルトのインストール先も新しい場所に更新されました
 - Windows: C:\Program Files\NV5
 - Linux: /usr/local/nv5
 - Mac: /Applications/NV5

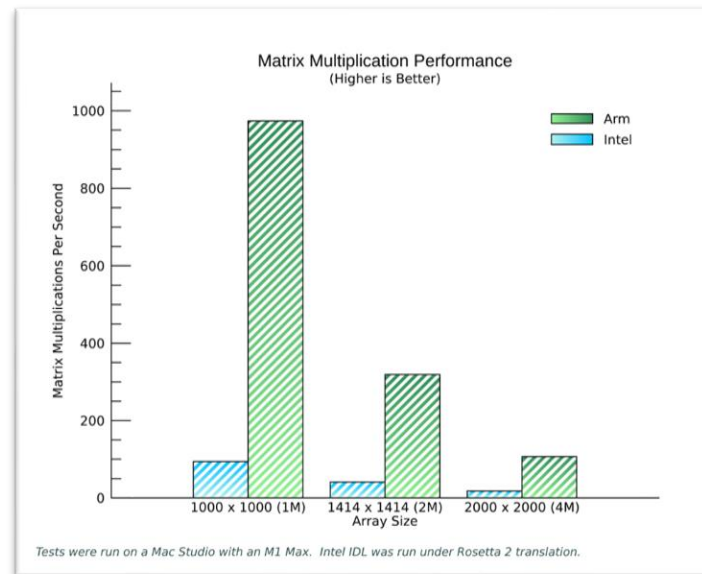


- Apple Silicon Macをサポート

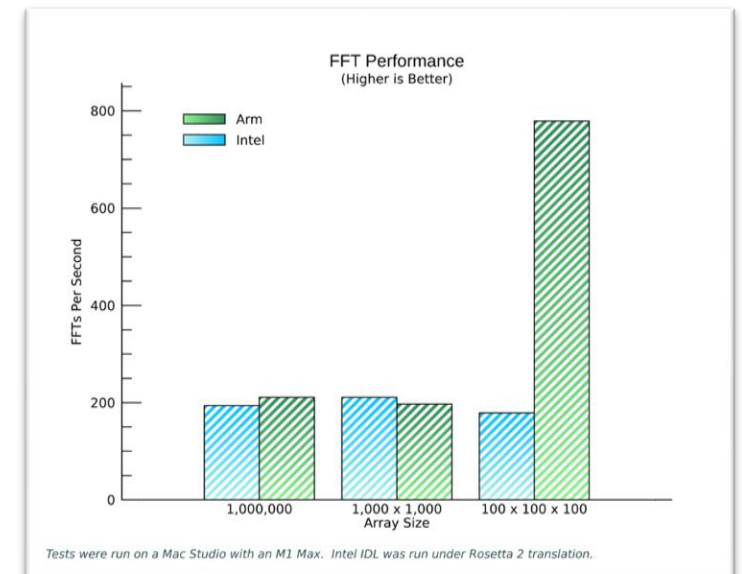
- IDLはAppleシリコンMac上でネイティブに実行できるようになりました。これにより、IDLはAppleのMシリーズチップを最大限に活用できるようになります。
- AppleシリコンMac上のAppleのRosettaトランスレータでIntelチップ用に構築されたIDLを実行する場合と比較して、パフォーマンスが向上します。
- IntelビルドのIDLは、Intelハードウェアを搭載したMacで引き続き利用できます。



File I/O



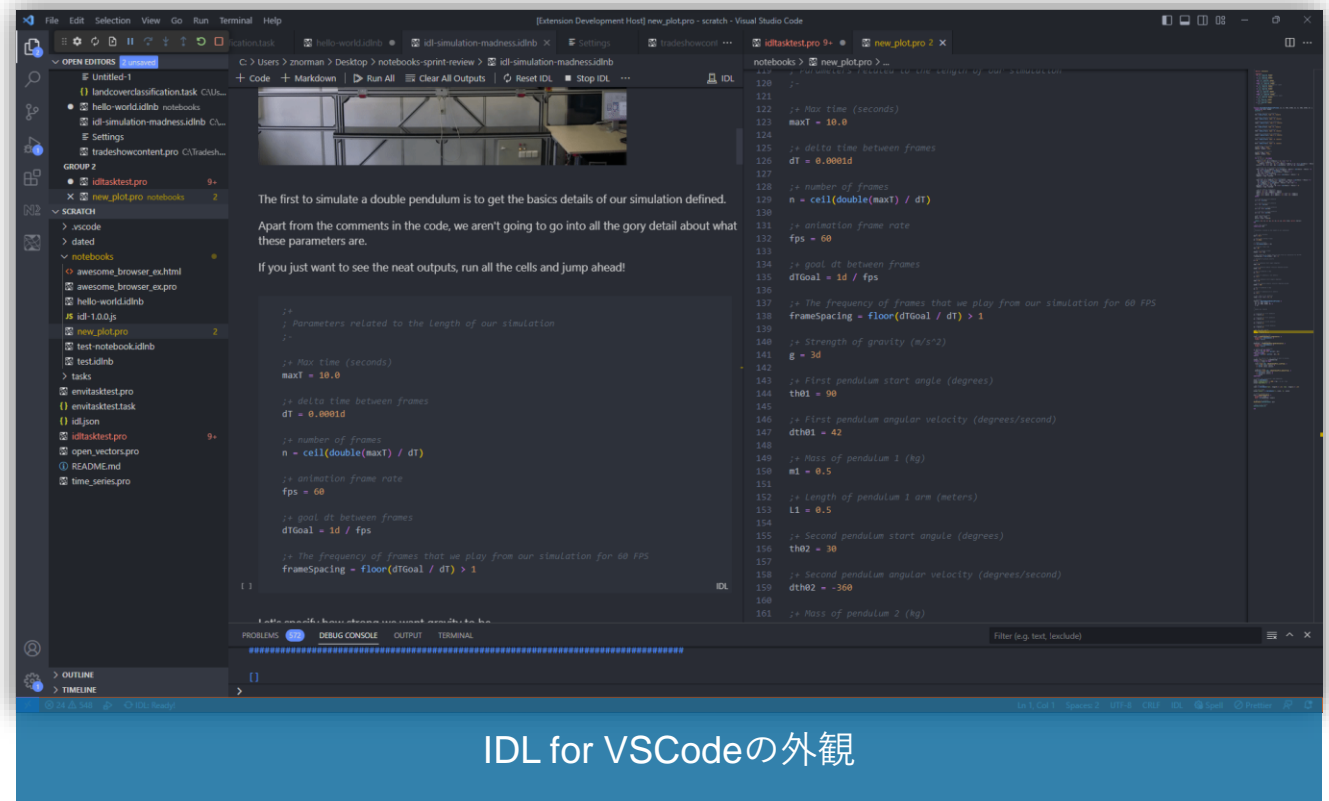
行列積



FFT演算

- IDL には、Visual Studio Code 内で無料で利用できる、最新の開発環境が追加されました。

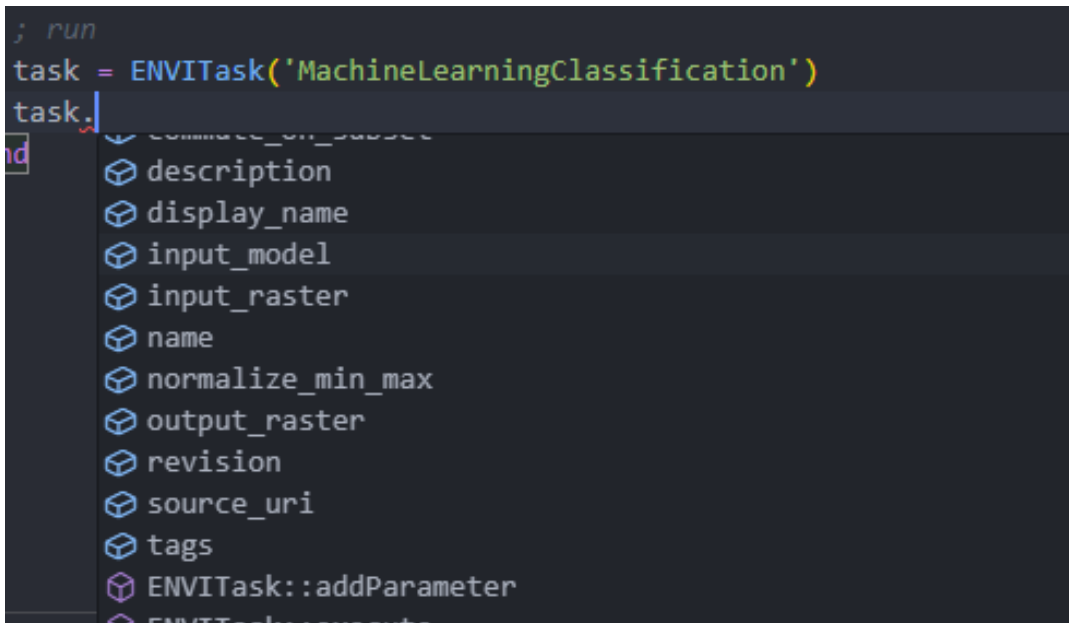
- ✓ シームレスなオートコンプリートのための基本的な IDL タイプ検出機能
- ✓ コードドキュメントのフォーマット、追加、更新を自動化
- ✓ ルーチン、メソッド、キーワード、変数のオートコンプリート機能
- ✓ ホバーヘルプには、コード例を含む完全な IDL ドキュメントが含まれます。
- ✓ 問題検出機能。コンパイルせずにコード内の 100 を超える問題を検出します。
- ✓ ブレークポイントを備えた統合デバッガー、または VS Code ターミナル内でコマンドライン IDL を実行します。



IDL for VSCodeの外観

- IDL には、Visual Studio Code 内で無料で利用できる、最新の開発環境が追加されました。

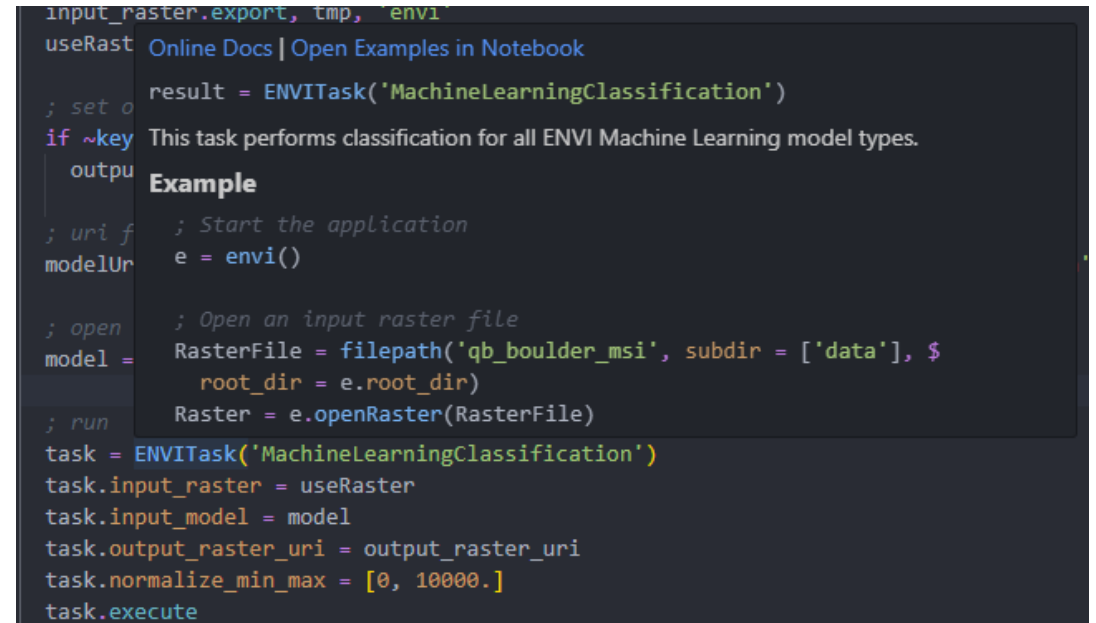
```
; run
task = ENVITask('MachineLearningClassification')
task.
nd
```



自動補完機能の例:

ENVI Task「MachineLearningClassification」のプロパティとメソッドが自動的に補完されている

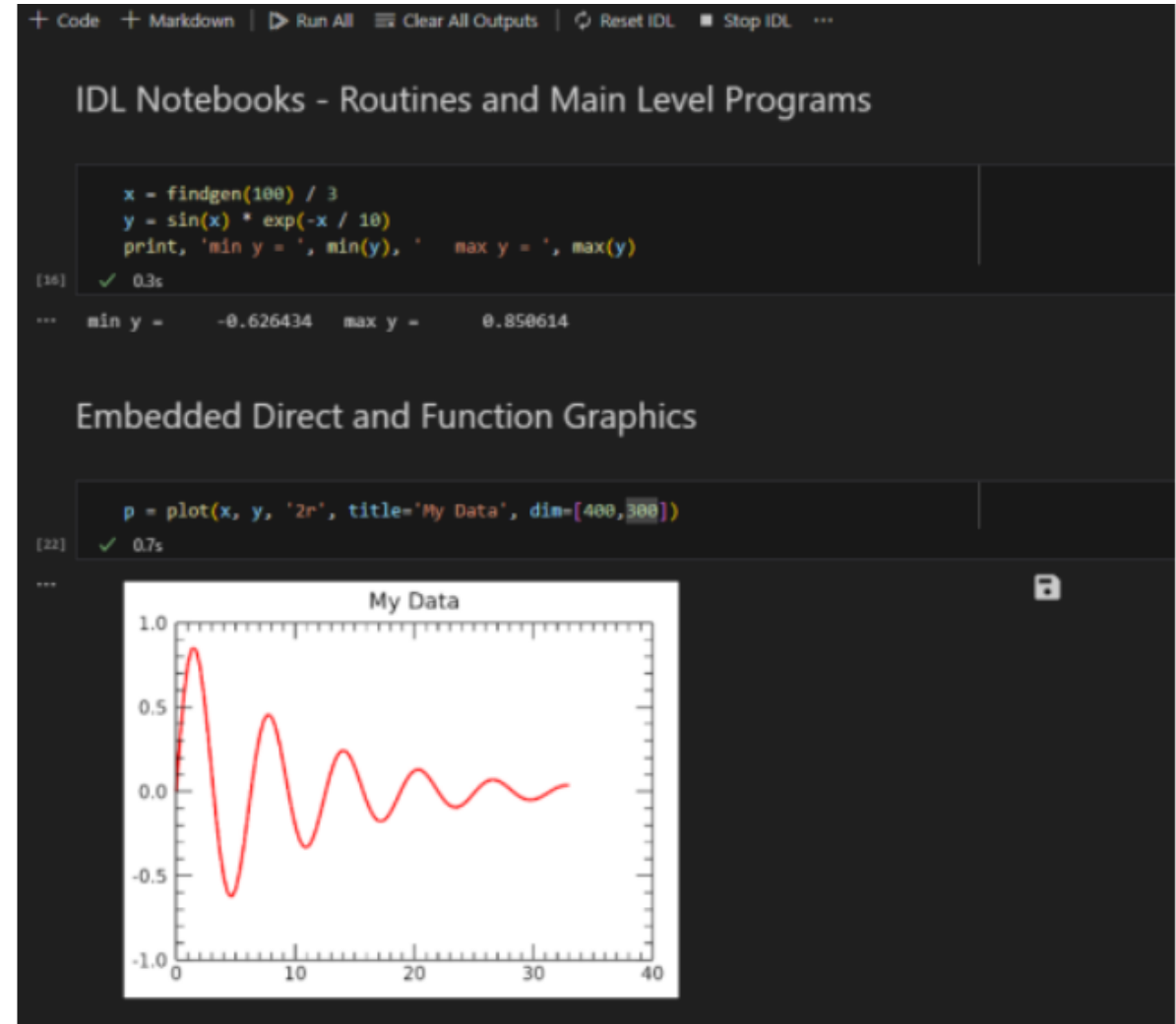
```
input_raster.export, tmp, 'envi'
useRaster Online Docs | Open Examples in Notebook
result = ENVITask('MachineLearningClassification')
; set o
if ~key This task performs classification for all ENVI Machine Learning model types.
output Example
; uri f ; Start the application
modelUr e = envi()
; open ; Open an input raster file
model = RasterFile = filepath('qb_boulder_msi', subdir = ['data'], $
root_dir = e.root_dir)
; run Raster = e.openRaster(RasterFile)
task = ENVITask('MachineLearningClassification')
task.input_raster = useRaster
task.input_model = model
task.output_raster_uri = output_raster_uri
task.normalize_min_max = [0, 10000.]
task.execute
```



サンプルスクリプト付きのホバーヘルプ機能

• IDL Notebook

- ネイティブ IDL Notebook 形式と開発者環境が追加されました。これは、IDL for VSCode 拡張機能を介して無料で入手できます。
- IDL Notebooks は完全にネイティブであり、Jupyter に基づいておらず、Python も必要ありません。
- テキスト マークアップとコードを 1 か所でキャプチャするための使いやすい形式です。
- ディスク上にファイルを作成してコマンドラインから実行する場合と比べて、より簡単です。



The screenshot displays the IDL Notebook interface. At the top, there are controls for running code: '+ Code', '+ Markdown', 'Run All', 'Clear All Outputs', 'Reset IDL', and 'Stop IDL'. The main area is titled 'IDL Notebooks - Routines and Main Level Programs'. It shows a code cell with the following IDL code:

```
x = findgen(100) / 3
y = sin(x) * exp(-x / 10)
print, 'min y = ', min(y), ' max y = ', max(y)
```

The code is executed, and the output is displayed below it:

```
min y = -0.626434 max y = 0.850614
```

Below this, there is another section titled 'Embedded Direct and Function Graphics'. It shows a code cell with the following IDL code:

```
p = plot(x, y, '2r', title='My Data', dim=[400, 300])
```

The code is executed, and a plot is displayed below it. The plot is titled 'My Data' and shows a red line graph of the function $y = \sin(x) \cdot \exp(-x/10)$ over the range $x \in [0, 40]$. The y-axis ranges from -1.0 to 1.0. The plot shows a damped sine wave starting at (0,0), reaching a peak of approximately 0.85 at $x \approx 3$, and then oscillating with decreasing amplitude towards zero.

- HttpRequest クラス

- いままでのIDLNetURLを発展させ、サーバーに対する Get、Post、Put、および Delete リクエストを簡単に実行できる新しいHttpRequestクラスが追加されました。
- 新しいクラスには、マルチパートフォームの投稿、認証、カスタムヘッダー、およびアップロードまたはダウンロード中のコールバック (リクエストをキャンセルするオプション付き) のサポートが含まれています。

```
; data that we want to send, including a file
multipart = dictionary('key1', 'value1', $
  'key2', {file: 'c:/image.jpg', mimetype: 'image/jpeg'})

; post our form!
response = HttpRequest.Post('https://httpbin.org/post', multipart = multipart)

; print information about the response
print, `status_code = ${response.status_code}`
print, response.json(), /implied
```

```
; parameters for a get request
params = hash('key1', '2+2', 'key2', 'with spaces ')

; use the "Get" statis method
response = HttpRequest.Get('http://localhost:3000', params=params, /escape)

; print information about our response
print, response.url
print, `status_code = ${response.status_code}`
help, response.text
```

HttpRequest API の例:

上: マルチパートフォームのPOST処理

下: リクエストのGET処理

- 文字列および変数にDupメソッドが追加されました
 - スカラー値を複製し、指定された次元の配列を返します
- MATRIX_MULTIPLY関数にNANキーワードが追加されました
 - 入力データに含まれるNaNや無限(発散)をゼロとして扱います
- 変数の属性値にTYPESIZEが追加されました
 - Float型であれば4を返すなど、変数とバイト間の変換を行う場合に便利です。

- ARROW 関数は破線と点線のスタイルをサポートしました
- MAP グラフィックス機能は、マップの作成時に、フォント サイズに基づいて、より適切なデフォルトのマージンを使用するようになりました。
- MODIFYCT でカラー テーブルを削除できるようになりました。
- READ_TIFF ルーチンと WRITE_TIFF ルーチンは、Zstandard (zstd) 圧縮を使用して読み書きできるようになりました。
- IDL Python Bridge が Python 3.11 をサポートするようになりました

NV5 Geospatial株式会社
技術サポート
support_jp@nv5.com