

# IDL 9.1 新機能紹介

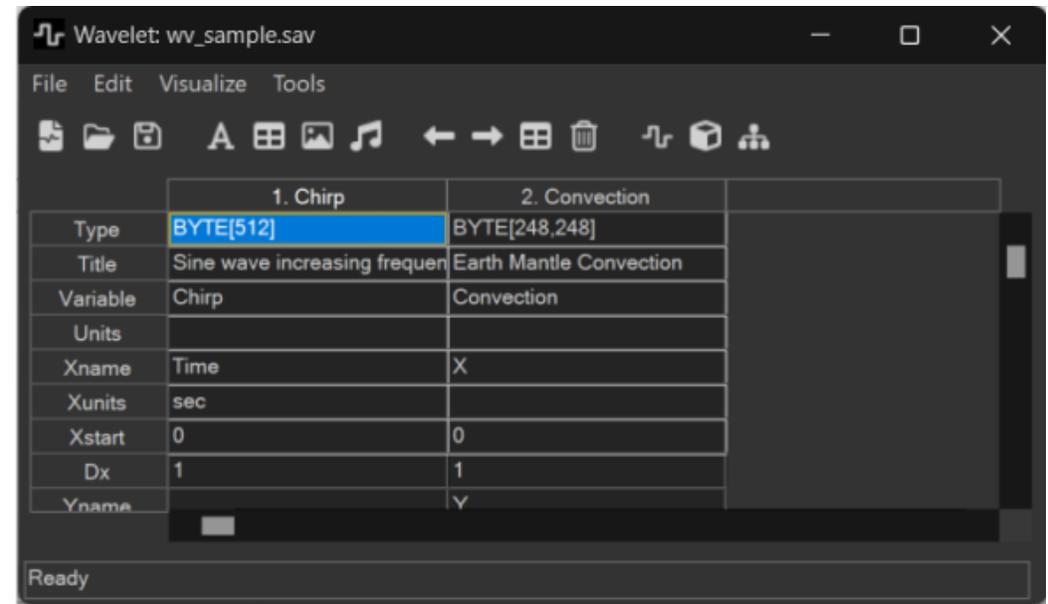
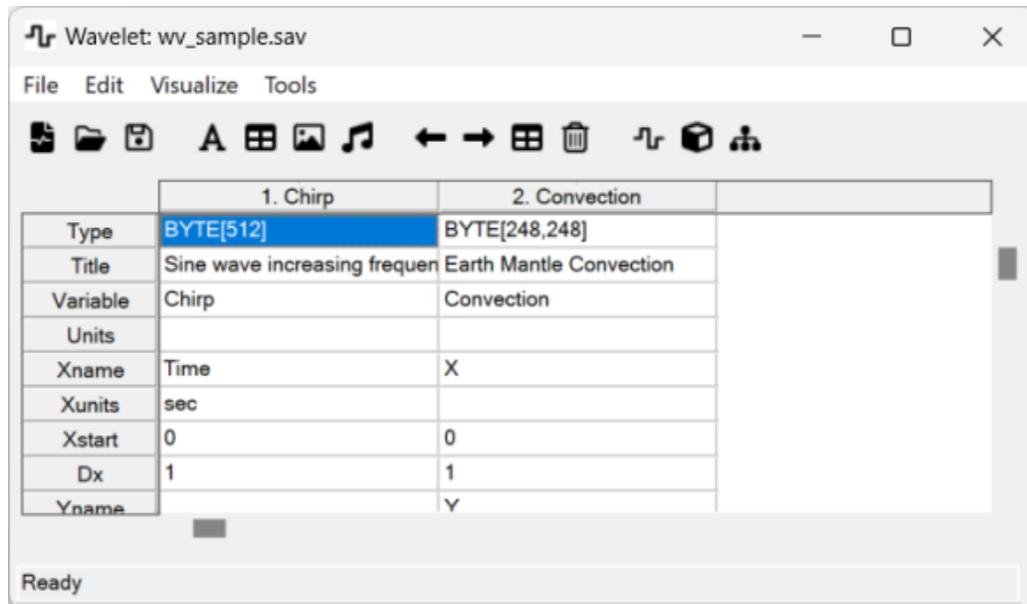
NV5 Geospatial 株式会社

N|V|5

- 新機能
- 機能向上

## Windowsのウィジェットにダークモードが追加されました

- 新しい環境変数 IDL\_THEMEを設定することでダークモードが利用できます（※Windowsのみ）
- LinuxとMacでは、この環境設定は無視され、常にライトモードが使用されます。
- ダイレクトグラフィックウィンドウとオブジェクトグラフィックウィンドウ内のグラフィックは影響を受けません。



## 球面調和変換の関数が導入されました

- 球面調和変換は、全球データセットの波浪パターンを分析するのに便利なツールで、気象学や海洋学、地磁気や太陽磁場の研究で広く使われています。
- 球面調和変換を球面座標グリッド上で順変換する SPHERE\_HARM\_FORWARD 関数と、それに対応した逆変換の SPHERE\_HARM\_INVERSE です。
- この順変換/逆変換のための関数に加えて、閾値より長い波数を保持するためのスペクトルフィルタとして有用な SPHERE\_HARM\_FILTER および SPHERE\_HARM\_KERNEL 関数も導入されています。

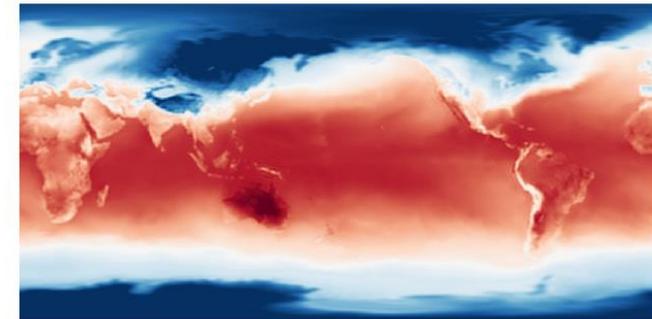
サンプル:

上: 1884年12月1日の気温の全球データ

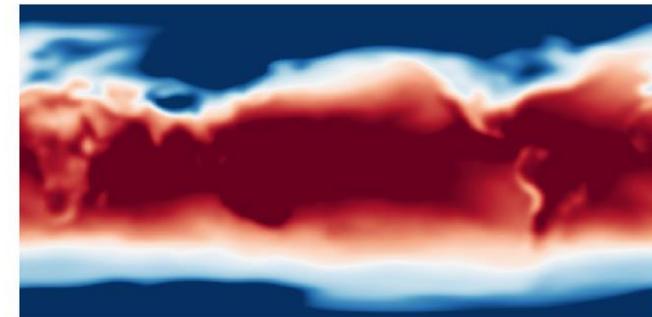
中: 球面調和変換を使ってデータをフィルタリングし、大気中の低周波の波を保存したもの

下: 低周波データと元データの差

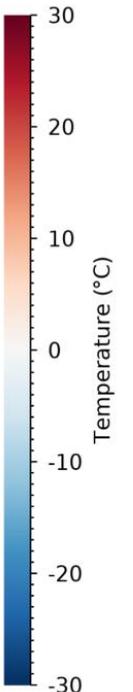
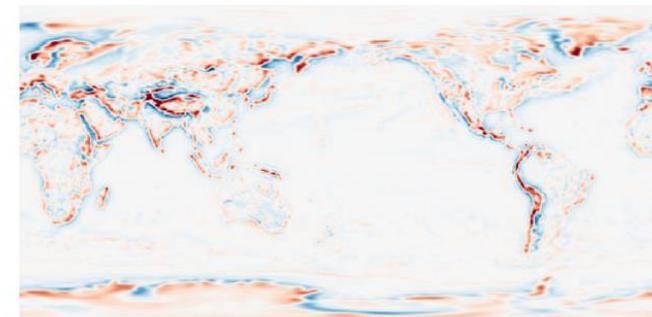
Original Data (2m Air Temperature) 1 Dec 1884



Filtered Data, T42 truncation



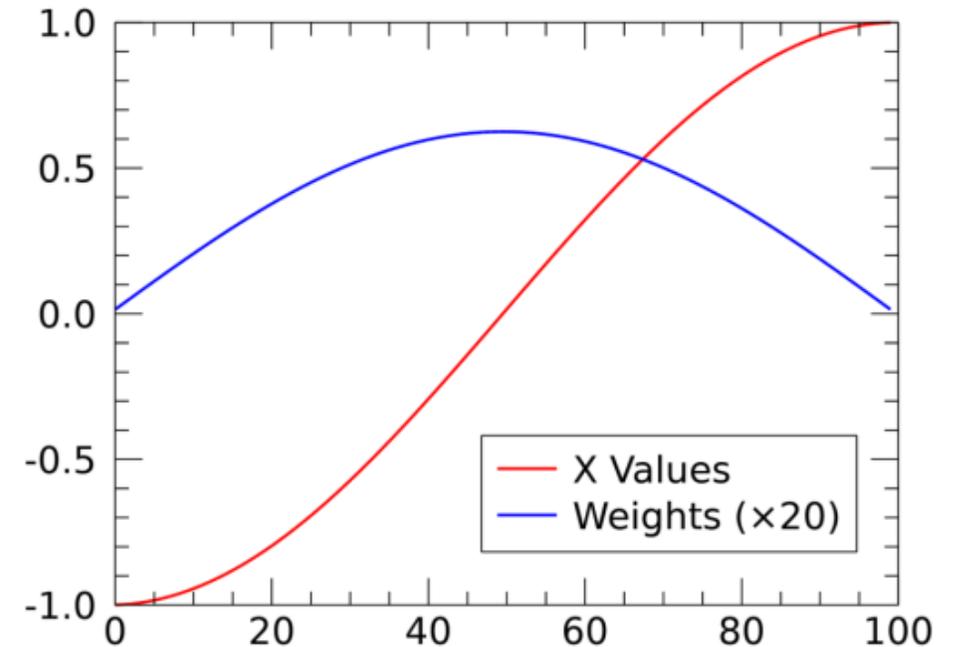
|Original - T42| x2



## GAUSS\_QUAD\_LEGENDRE関数が追加されました

- ガウス積分のx値 (横座標)を含む配列を返します。
- ここで、x値はN次ルジャンドル多項式の根です。

$$\int_{x_1}^{x_2} f(x)dx = \sum_{j=1}^N w_j f(x_j)$$
$$w_j = \frac{2}{(1 - x_j^2)[P'_N(x_j)]^2}$$



```
IDL> x = gauss_quad_legendre(-1, 1, 100, weights=w)
IDL> p1 = plot(x, 'r2', font_size=18, name='X Values')
IDL> p2 = plot(w * 20, 'b2', /overplot, name='Weights ($¥times$20)')
IDL> l = legend(target=[p1, p2], position=[0.88, 0.35], font_size=18)
```

## 半精度浮動小数点数の読み取りと変換に対応しました

- 半精度浮動小数は 16 ビットの長さ定義で、これは単精度浮動小数(32 ビット)の半分です。
- IDL9.1では 16 ビット浮動小数点数をネイティブサポートはしていませんが、float16 数値の読み取りと変換ができるようになりました。
  - READ\_BINARY関数には新しい FLOAT16 キーワードがあり、ファイルから float16 数値を自動的に読み取り、32 ビット浮動小数点数に変換します。
  - 新しく追加されたFLOAT16\_DECODE関数とFLOAT16\_ENCODE関数を使用して、16 ビット符号なし整数 (float16 形式) と 32 ビット浮動小数点数を相互に変換できます。

## コマンドラインプロセッサ CLI\_Progress が新たに実装されました

```
cli_progress.initialize
for i = 0, 100 do begin
    wait, 0.02
    ; your code here
    cli_progress.update, i
endfor
; [CONSOLE OUTPUT]
; 49% [#####-----]
```

## IDL\_HASHVAR 関数が導入されました

- IDL\_HASHVAR関数は、指定された変数のハッシュ コード値を返します。
- ハッシュコード値は、Bob Jenkins の One-At-A-Time ハッシュを使用して計算され、32 ビットのハッシュ コードが生成されます。
- この関数は、以下のような用途での利用を想定しています:
  - 任意の値を均一に分布する32ビットのハッシュコードに変換する場合
  - 2つの文字列が同一であるかどうかを判断するような場合 (文字列のわずかな違いでも完全に異なるハッシュ値になるため)
- この関数は暗号化用に設計されていないため、暗号化の目的には使用しないでください。

```
print, idl_hashvar(123b), idl_hashvar(123.0), idl_hashvar(123uLL)
; output > 3044569485 3044569485 3044569485
; 同じ値は別の型でも同じハッシュ値になる
```

```
print, idl_hashvar('Keanu Reeves'), idl_hashvar('Keanu Jeeves')
; output >
; 3160331264
; 3683530696
; 一文字違いでもハッシュ値は全く異なる
```

## 新しいSVGアイコンが提供されました

- IDL9.1には、Font Awesome プロジェクトからの 2000 を超える新しいアイコンが付属しています。
  - SVG\_ICON\_BROWSER プロシージャで、すべてのアイコンを表示できます。
- これらのアイコンはすべて Creative Commons Attribution 4.0 International License に基づいて配布されており、独自のアプリケーションで自由に使用できます。



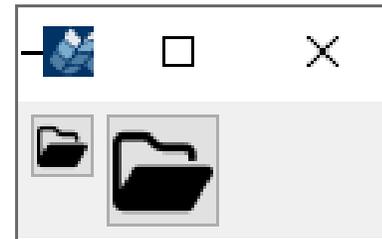
## 新しいSVGアイコンが提供されました

- オプションの出力サイズ、背景色、前景色を使用して SVG ファイルをビットマップイメージにレンダリングできる `RENDER_SVG` 関数が新たに追加されました。
- IDL インストールフォルダ内の SVG ファイルを読み取り、ウィジェットボタンまたはウィジェットツリーアイコンとして使用するための `RENDER_ICON` 関数が新たに追加されました。

```
file = filepath('folder-open.svg', subdir=['resource', 'bitmaps', 'svg', 'regular'])  
img = RENDER_SVG(file, width=128, height=128, background=0xC0C0C0, foreground=0xFF0000)  
i = image(img)
```



```
w = widget_base(/row)  
icon = RENDER_ICON("folder-open")  
w1 = widget_base(w)  
wb = widget_button(w1, value=icon, /bitmap)  
icon = RENDER_ICON("folder-open", width=32,  
height=32)  
w1 = widget_base(w)  
wb = widget_button(w1, value=icon, /bitmap)  
widget_control, w, /realize
```



## Visual Studio CodeのIDL拡張機能がアップデートされました

- VS Code 拡張機能ページから簡単にダウンロードしてインストールできます。
- 拡張機能には次のような機能があります:
  - 変数の型検出、自動補完、クロマコーディング、問題レポート、コードフォーマットなどの機能を備えたフル機能のIDLコードエディター機能
  - 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 実行機能
  - 埋め込みグラフィックと PDF への保存を備えた IDL ノートブックスタイルのインターフェイスをサポート
  - カスタマイズ可能なテーマと色設定
  - 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ています。
  - SAVE ファイルまたは DLM 内のルーチンのルーチン定義ドキュメントを作成します。
  - ネイティブの多言語サポート

## Visual Studio CodeのIDL拡張機能がアップデートされました

The screenshot displays the Visual Studio Code interface with the IDL extension. On the left, the 'IDL ACTIONS AND COMMANDS' sidebar is visible, listing various actions like 'Start a session of IDL', 'Compile PRO file', and 'Run PRO file'. Below this, the 'Notebooks' section is expanded, showing options for creating and managing IDL notebooks. The main editor area shows the source code for 'wv\_applet.pro', which includes comments and code for defining wavelet functions and handling device-specific settings. The right sidebar shows a notebook titled 'Hello World: IDL!'. The notebook contains two cells: the first cell executes the command 'print, "Hello, World!'" and outputs 'Hello, World!'; the second cell executes 'c = contour(/test)' and displays a contour plot of a 2D function.

## HDF5の空の配列を処理できるようになりました

- H5D\_READおよびH5\_GETDATAルーチンは、長さがゼロの配列を持つ H5T\_VLEN データ型を処理できるようになりました。
- このような場合、結果は !NULL 変数へのポインターになります。

## HttpRequestのプログレスバーをサポートしました

- IDL9.0から導入されているHttpRequestクラスのGet、PostおよびPut メソッドにオプションとしてプログレスバーが追加されました。
- この進行状況バーは、IDL9.1からの新機能、CLI\_Progressオブジェクトを使用します。

## 凡例を表示するLEGEND関数にFILL\_COLORキーワードが追加されました

## JSONファイルのNaNおよびInfinity値をサポートします

- IDLのJSON\_PARSE関数は、配列の末尾のコンマと、特殊なNaN、Inf、およびInfinity (引用符なし) 値の両方をサポートするようになりました。
  - これらはすべてJSON仕様への非標準の追加ですが、PythonやJSONを生成するその他のツールとの相互運用性が向上します。
- JSON\_SERIALIZE関数に新しいNANINF\_LITERALキーワードが追加されました。
  - NANINF\_LITERALキーワードを設定すると、NaN値とInfinity値が引用符なしで出力されます。(従来は引用符で囲まれた文字列として出力されていました)
  - これにより、JSON\_PARSEやPythonなどの他のツールはこれらの値を理解し、浮動小数点NaN値とInfinity値を返すことができます。

## StringオブジェクトのReplaceメソッドにCount引数が追加されました

- この引数を使用すると、置換処理を行う際の一致数を細かく指定できます。

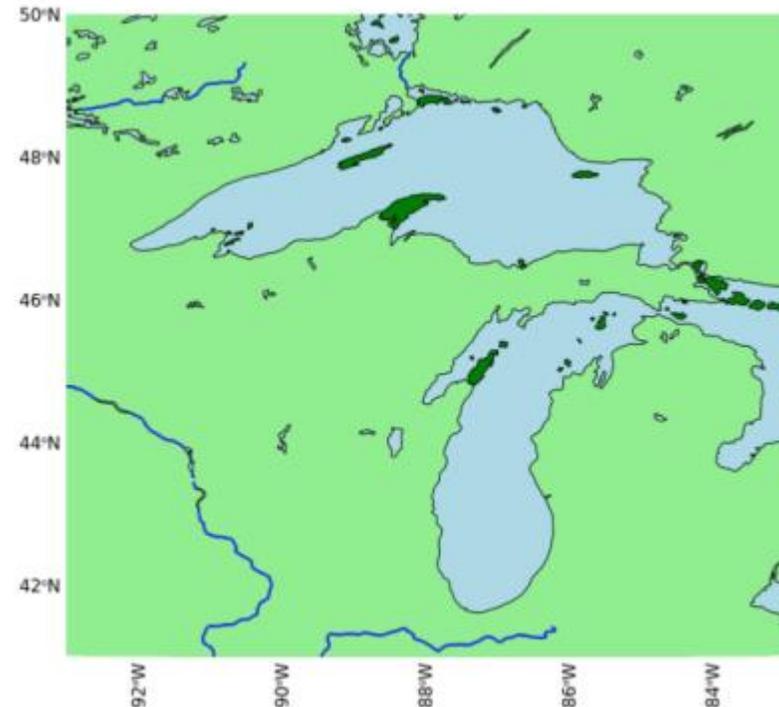
```
str = '.....'  
print, str.replace('.', '*', 3)  
; [output] ***.....
```

## MapContinents関数のシェープファイルが更新されました

- MapContinentsグラフィックス機能は、大陸、国、湖、島、河川の最新のシェープファイルで更新されました。
- これには、大陸、湖、河川の新しい高解像度バージョンが含まれます。

```
m0 = Map('Geographic', limit=[41, -93, 50, -83], $
margin=[0.1, 0.1, 0.1, 0.05])
m0.mapgrid.label_position = 0
m0.mapgrid.linestyle = "none"
m0.mapgrid.grid_longitude = 2
m0.mapgrid.grid_latitude = 2

m1 = MapContinents(/hires, fill_color="light green")
m2 = MapContinents(/lakes, /hires, fill_color='light blue',
linestyle=0)
m3 = MapContinents(/islands, fill_color='green')
m4 = MapContinents(/rivers, /hires, color='blue', thick = 2)
```



## ウィジェットが複数行のツールチップをサポートします

- ツールチップとは、用語やフォーム等の対象にマウスオーバーした際に表示される注釈や補足情報を指します。
- TOOLTIPキーワードを設定するときに改行 ¥n 文字を埋め込むだけで実現可能です。

## PRINT/PRINTFプロシージャにNEWLINEキーワードが追加されました

- NEWLINEキーワード(デフォルト値1)に対して0を設定すると、改行が抑制され、PRINT を複数回呼び出して、すべての出力を同じ行に表示できるようになります。

## シェープファイルの機能が強化されました

- IDLffShapeクラスで使用されるShapefileライブラリが最新バージョンに更新されました。
- さらに、IDLffShapeはDateおよびLogical属性タイプを処理できるようになりました。

## TS\_SMOOTH関数にNaNキーワードが追加されました

- TS\_SMOOTH関数ではN個の時系列要素の中心、後方、または前方の移動平均が計算できます。
- NaNキーワードを設定すると、入力データ内のIEEE浮動小数点値NaNまたはInfinityの発生をチェックし、これらを欠落データとして扱うようになります。

## WIDGET\_LABELとWIDGET\_TEXTのテキスト/背景色の変更ができます

- FOREGROUND\_COLORとBACKGROUND\_COLORキーワードが追加されました。

```
pro widtext_ex_change, ev
  common _widtext, w1, w2
  widget_control, w1, foreground_color=bytsc1(randomu(s,3)), background_color=bytsc1(randomu(s,3))
  widget_control, w2, foreground_color=bytsc1(randomu(s,3)), background_color=bytsc1(randomu(s,3))
end

pro widtext_ex
  common _widtext, w1, w2
  w = widget_base(xsize = 200, /column)
  w0 = widget_base(w, /row)
  f = 'Segoe*24'
  w1 = widget_label(w0, value='Label', foreground_color=[0,200,0], background_color=[255,255,0],font=f, scr_xsize=100)
  w2 = widget_text(w0, value='My Text', /editable, foreground_color=[200,0,0], background_color=[0,255,255], font=f)
  w0 = widget_base(w, /row)
  wbutton = widget_button(w0, value='Change', event_pro = 'widtext_ex_change', font=f)
  widget_control, w, /realize
  xmanager, 'widtext_ex', w, /no_block
end
```



WIDGET\_BROWSER関数がmacOSでも動作をサポートします

WIDGET\_PROPERTY SHEET関数に、ヘッダを表示/非表示できるTOP\_HEADERキーワードが追加されました

WIDGET\_TEXTに、ALIGN\_\*といった配置に関するキーワードが6つ追加されました

WIDGET\_TEXTに、周囲のボーダーを表示/非表示させるWIDGET\_TEXT FRAMEキーワードが追加されました(※Windowsのみ対応)

WRITE\_PNG 関数に、PNG 出力の圧縮率を制御する新しいキーワード COMPRESSION が追加されました。

データベース管理システムとの連携機能であるDataminerモジュールが、デフォルトでIDLとともにインストールされるようになりました(※利用には引き続きライセンスが必要です)

NV5 Geospatial株式会社  
技術サポート  
support\_jp@nv5.com