



# Automate, Scale, and Accelerate: Inside ENVI 6.3 and IDL 9.3

September 2026

**N|V|5**

# Before we begin...



- The webinar is being recorded
  - Slides & on-demand version of the webinar will be available within a few business days



- Questions?
  - Please post any questions in the Questions box

# Today's Speaker



**Zach Norman**

*Product Manager*

[zachary.norman@nv5.com](mailto:zachary.norman@nv5.com)

# Agenda

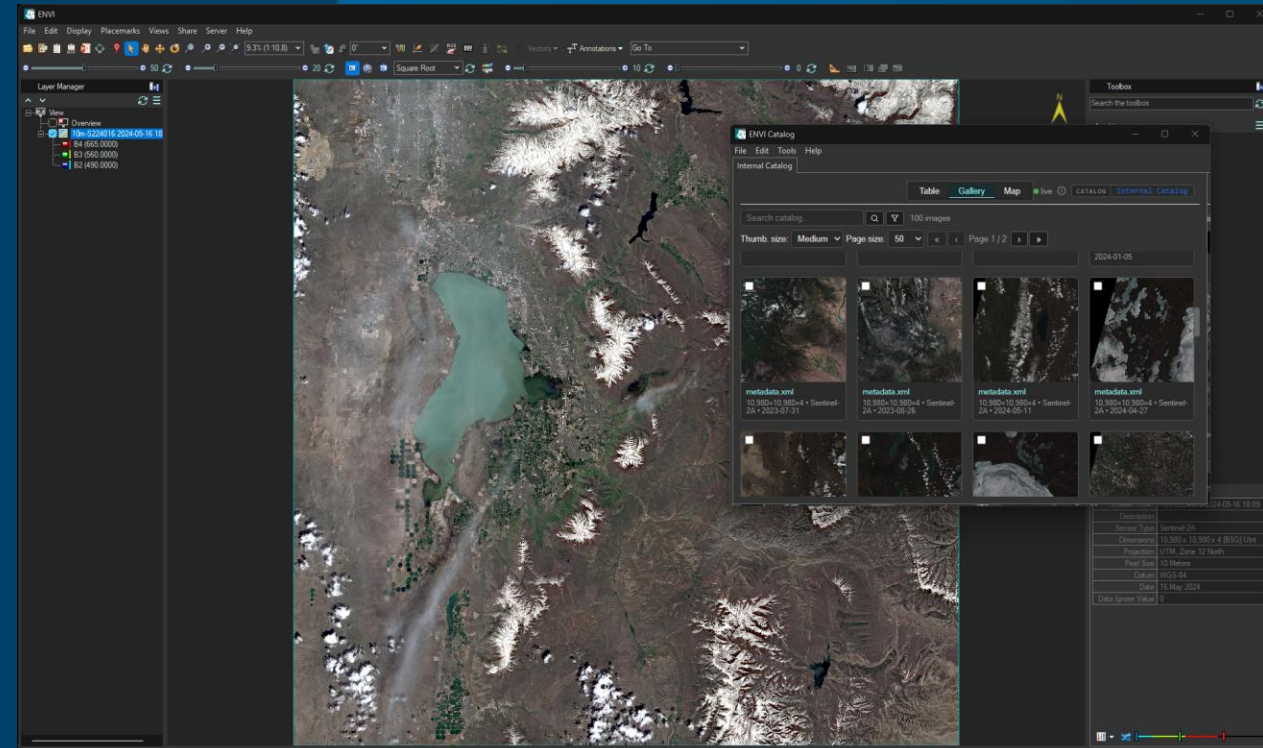


- ENVI 6.3
  - Release Highlights
  - Demo: Desktop Automation and ENVI Catalog
- IDL 9.3
  - Release Highlights
  - Demo: Embedded Python and new Graphics Exports in an IDL Notebook
- IDL Agent and ENVI Agent
  - Recent Changes
- Full release notes are online
  - [IDL 9.3](#)
  - [ENVI 6.3](#)

# ENVI Catalog

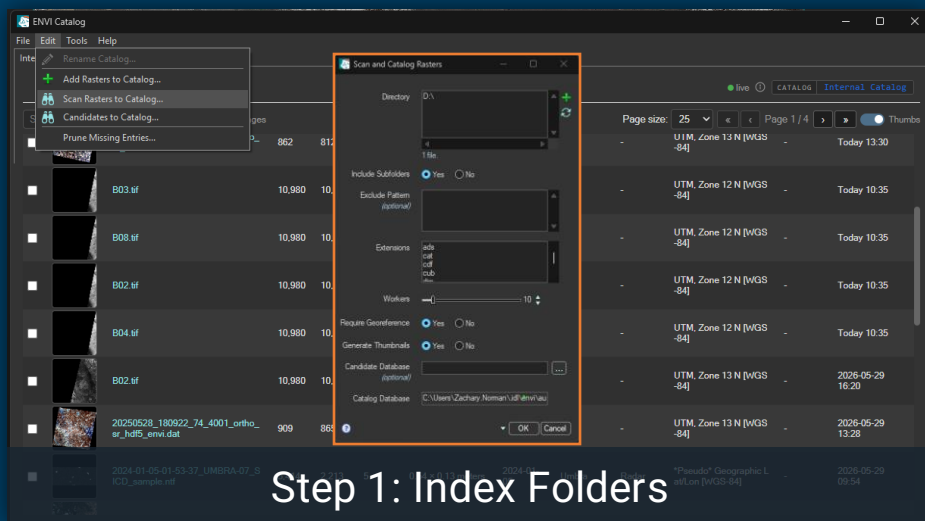


- A new addition to ENVI Pro that allows you to index and explore data that exists locally on your system
- Use cases
  - Help individuals keep track of the datasets they accumulate over time as they work on new remote sensing projects
  - Make it easier for users who share hardware for shift-work track and understand what data products are available

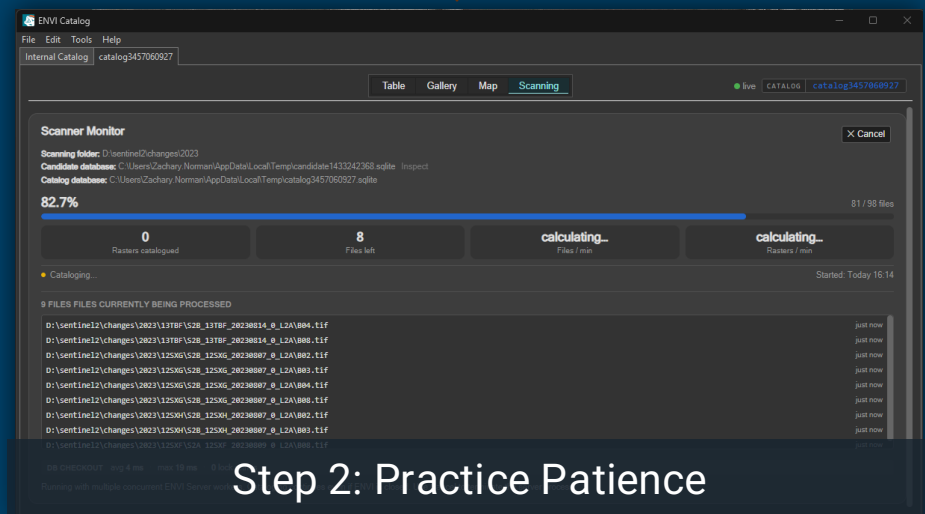


ENVI Catalog helping NV5 Product Managers track 100+ Sentinel 2 scenes that are used for machine learning demonstrations

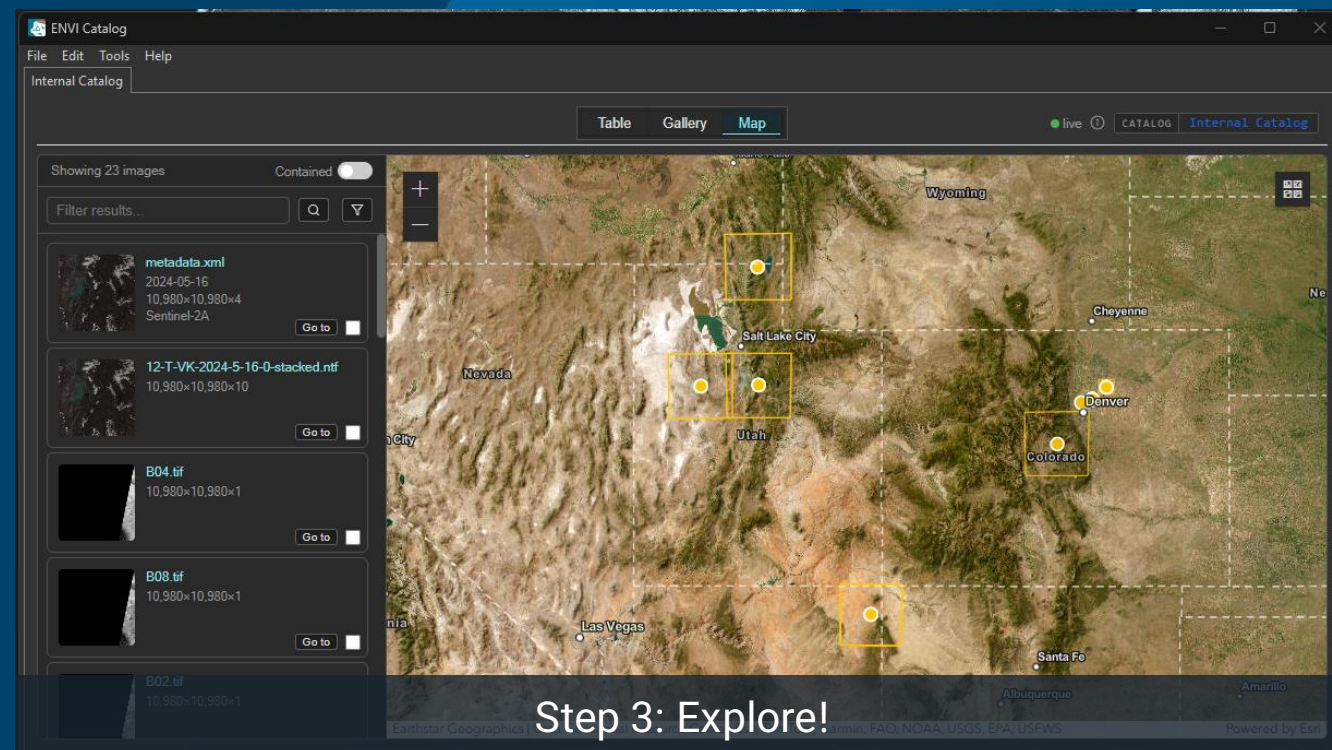
# ENVI Catalog: General Workflow



Step 1: Index Folders



Step 2: Practice Patience



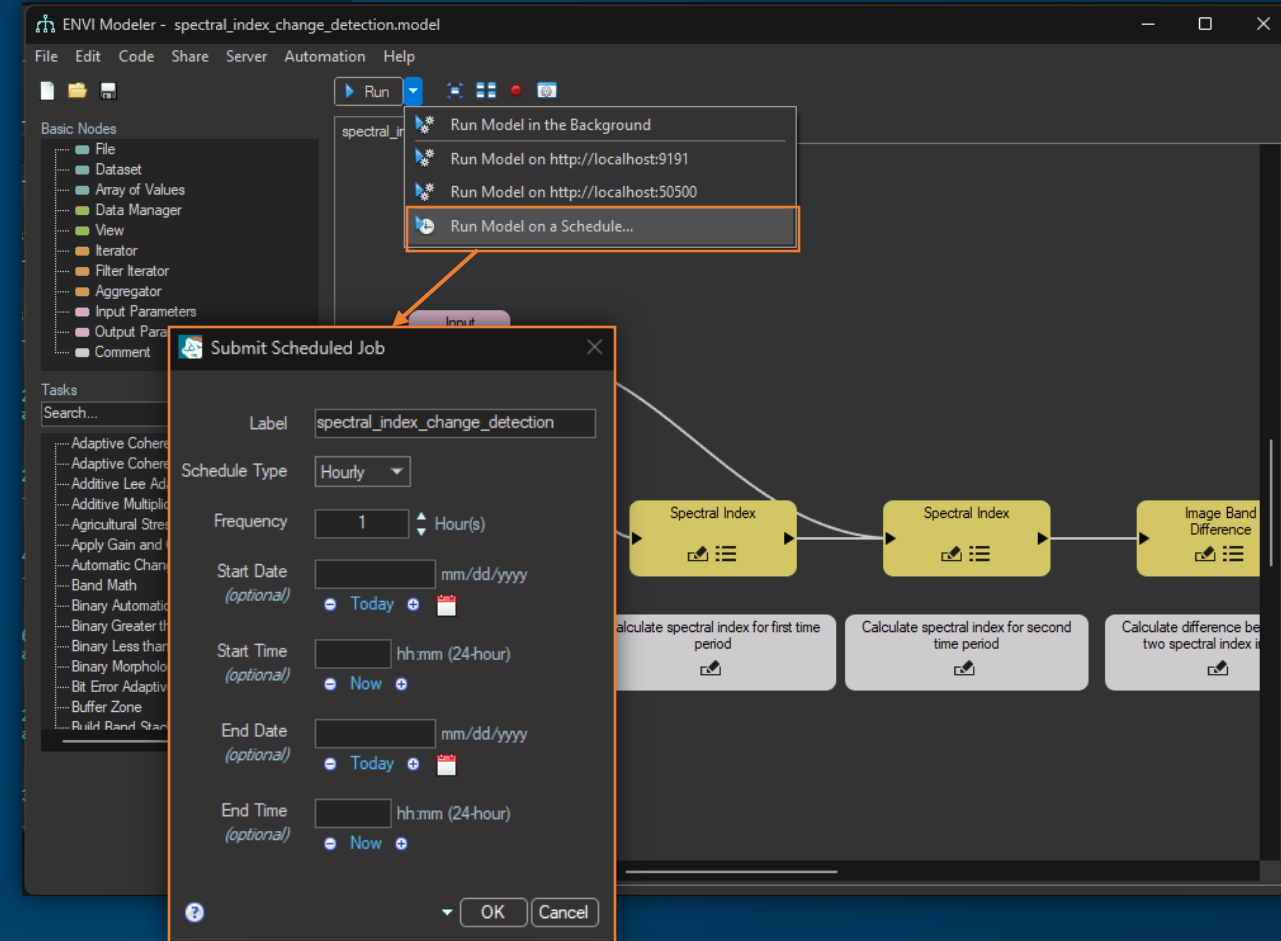
Step 3: Explore!

Note: Indexing is something you trigger at the end of the day for large data stores

# Desktop Automation



- New capability that allows users to run processing in ENVI on a schedule
- Use cases
  - Process all new data that appears in a folder with a ship detection deep learning model
  - At the end of each day, process all newly downloaded data using a vegetation analysis workflow

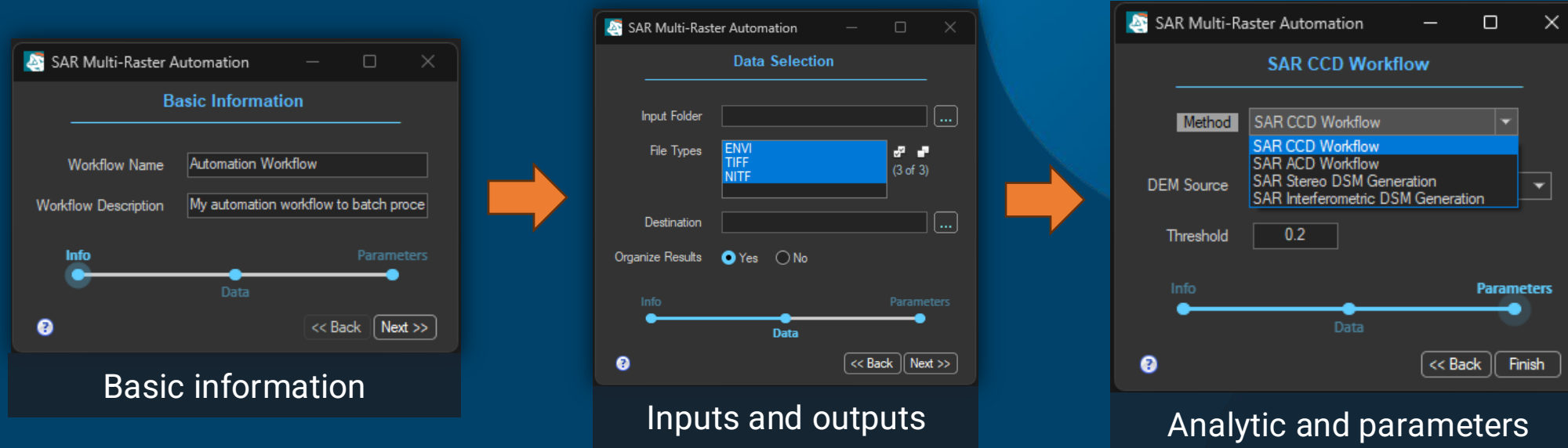


# Desktop Automation: Extensible



Desktop Automation can be fine-tuned to present clean, easy-to-use workflows for users without the need to work with the ENVI Modeler

Here is an example of a custom workflow that detects change between SAR datasets



# ENVI 6.3 Demo!

- Set up a Desktop Automation workflow
- Batch process data
- Ingest data into the ENVI Catalog



# IDL 9.3

- Next, we will cover the new features included in IDL 9.3
- Check out the [release notes](#) for a full list of features
- Highlights:
  - New syntax for strings and Lambda functions
  - Increased array dimensions
  - BigFloat data type for up-to 1000 digits of precision
  - Embedded Python updated and IDL Agent can now help manage your environment

# New Lambda Function Syntax

Array Filter, Map, NestedMap, Reduce  
Functions Now Work with Lists

```
a = ['Arr1', 'Arr2', 'Arr3']  
b = List('L1', 'L2', 'L3')  
result = a.Map(Lambda(x, y: `${x} + ${y}`), b)  
result
```

✓ 0.2s

IDL

```
Arr1 + L1  
Arr2 + L2  
Arr3 + L3
```

Access the keys for hash values in  
Lambda's in Hash::Map with the  
"/keys" keyword

```
var = Hash("key1", "A", "key2", "B")  
result = var.Map(Lambda('value,sep,key: key + sep + value'), "=", /keys)  
result
```

✓ 0.2s

IDL

```
{  
  "key1": "key1=A",  
  "key2": "key2=B"  
}
```

# IDL Arrays: 64 dimensions

- IDL's arrays now support up to 64 dimensions!
- Array creation and helper functions support have been updated
- Includes routines like:
  - Lindgen
  - Findgen
  - Min
  - Max
  - Total
  - FFT
  - CONVOL
  - All other array-based routines!

```
; Example with 10 dimensions (2x...x2x4)
a = lindgen(2, 2, 2, 2, 2, 2, 2, 2, 2, 4)
help, a
print, 'ndim = ', a.ndim
print, 'dim  = ', a.dim
print, 'max  = ', max(a)

; Get max value along 10th dimension
r = max(a, dimension = 10)
help, r
```

# New Type: BigFloat

- The BigFloat function lets you create floating point numbers with up to 1000 digits of precision
- Works seamlessly with BigInteger for high-precision calculations
- Useful for number theory research, algorithm verification, chaos theory, and orbit analysis

```
; Create big floats from numbers or strings
```

```
b = BigFloat(3.14d)
```

```
b = -BigFloat('3.14')
```

```
b = BigFloat(314, -2)
```

```
; Use big floats like any other variable to
```

```
; calculate the Rydberg constant
```

```
fine_struct = 1 / BigFloat('137.035999177')
```

```
bohr_radius = BigFloat('5.29177210544e-11')
```

```
rydberg = fine_struct / (4 * BigFloat.pi * bohr_radius)
```

```
print, 'Rydberg estimate (BigFloat):'
```

```
print, rydberg
```

```
; Special values through use of keywords
```

```
b_inf = BigFloat(/infinity)
```

```
b_ninf = BigFloat(/infinity, /negative)
```

```
b_nan = BigFloat(/nan)
```

```
print, b_inf, b_ninf, b_nan
```

# WebSocket Connections

- IDL now supports WebSocket!
- Enabled by the HTTP class, this makes it easy to connect to IDL and communicate from nearly any language or toolset
- Workflow supports:
  - Creating connections
  - Checking for messages
  - Sending messages

```
; Routine that polls the socket connection for new messages  
; and cleans up if our connection is closed  
;  
; See documentation for full example  
pro test_callsocket, id, ws  
    compile_opt idl2  
    on_error, 2  
  
    result = ws.receive()  
    if isa(result, 'byte') || strlen(result) gt 0 then begin  
        print, result  
        if (isa(result, 'string') && result.contains('Close')) then begin  
            ; Close our WebSocket connection  
            ws.send, /close  
            !null = Timer.cancel(id)  
        endif  
    endif  
end  
  
; Make a WebSocket connections  
ws = HttpRequest.WebSocket('ws://localhost:8765')  
  
; Set timer to handle async messages for IDL  
id = Timer.set(0.5, 'test_callsocket', ws, /repeat)  
  
; Send a message  
ws.send, 'Hello from IDL'
```

# Embedded Python Bridge

- Starting with IDL 9.2, IDL and ENVI now include a native Python installation
  - IDL 9.2 includes Python 3.13
  - IDL 9.3 includes Python 3.14
- The PyUtils class enables package management and helps return information about your Python session

```
; List installed Python packages:  
PyUtils.PipList  
  
; Install Python package:  
PyUtils.PipInstall, 'beautifulsoup4'  
  
; Print information about a specific package  
PyUtils.PipShow, 'beautifulsoup4'  
  
; Uninstall Python package:  
PyUtils.PipUninstall, 'beautifulsoup4'  
  
; Get information about which Python  
; is configured  
info = PyUtils.getCurrentPython()  
print, info, /implied_print
```

# IDL 9.3 Demo!

- Example notebook that uses the Embedded Python and exporting graphics using new formats



## **IDL Agent**

A programming partner that empowers you to autonomously write, run, and troubleshoot IDL code, simplifying developer workflows and accelerating discovery



## **ENVI Agent**

A geospatial partner for planning and executing workflows that keeps you in the driver's seat, allowing you to direct the analysis while it handles the technical heavy lifting

# IDL Agent has a New Skill: Embedded Python

- The embedded Python environment is now available to IDL Agent!
- The Agent has specialized instructions that allow it to access the Python included with IDL and ENVI
- The goal is to:
  - Help users install and work with Python packages more easily
  - Enable the agent to use Python more proactively with your IDL code

Can you install "beautifulsoup" in IDL's embedded Python environment?

I'll check the local IDL guidance first, then identify the embedded Python executable and install the package into that environment.

Read [idl.instructions.md](#)

IDL Agent: The relevant guidance points to the bundled Python resource, so I'm retrieving that setup detail before touching the environment.

Searched for tool usage in the project

Ran Search Resources – IDL for VSCode (MCP Server)

IDL Agent: The documented install path is `PyUtils.PipInstall`, `'beautifulsoup4'` inside IDL, which installs the package under the embedded interpreter. I'll run that through the active IDL session and then verify that `bs4` imports successfully.

Searched for tool usage in the project

Ran Execute IDL Code – IDL for VSCode (MCP Server)

Ran Execute IDL Code – IDL for VSCode (MCP Server)

IDL Agent: Installed `beautifulsoup4` in IDL's embedded Python environment.

Verified successfully with `bs4` version `4.15.0`.

# IDL Agent has New Tools

- IDL Agent has been outfitted with new tools to help it operate just like a user does when they manually debug and run IDL code
- New tools and their use:
  - **Control IDL Debugger** to manage breakpoints and step through code
  - **Get IDL State** to learn about variables, call stack, and more
  - **Query IDL Session** to run IDL commands at the current scope

Can you tell me what variables have been set in IDL?12:18 PM

Searched for tool usage in the project

Ran Get IDL State – IDL for VSCode (MCP Server)

Currently set variables in the IDL session:

| Name      | Type      | Value                    |
|-----------|-----------|--------------------------|
| bar       | String    | 'Don't call me Shirley!' |
| dataroot  | undefined | <Undefined>              |
| dem       | undefined | <Undefined>              |
| e         | envi      | (ENVI session object)    |
| foo       | Long      | 42                       |
| headless  | undefined | <Undefined>              |
| s1        | undefined | <Undefined>              |
| s2        | undefined | <Undefined>              |
| watermask | undefined | <Undefined>              |

Note several ( `dataroot`, `dem`, `headless`, `s1`, `s2`, `watermask` ) are declared but currently undefined (no value assigned).



## ENVI Agent: Now a SAR Expert



### ENVI API

IDL Agent enables you to write and run code that interacts with ENVI, to create new tools and solve problems with a paired programmer



### Image Analysis

Full suite of ENVI's processing tools for processing all kinds of imagery datasets



### SAR Expertise

Full suite of SARscape's processing tools, including specialized SAR workflows



## Questions and Discussion

**Zach Norman**

*Product Manager*

[zachary.norman@nv5.com](mailto:zachary.norman@nv5.com)

**Jodi Lee**

*Marketing Lead*

[jodi.lee@nv5.com](mailto:jodi.lee@nv5.com)

**NV5 Geospatial**

[www.NV5GeospatialSoftware.com](http://www.NV5GeospatialSoftware.com)

[geospatialinfo@nv5.com](mailto:geospatialinfo@nv5.com)

303-786-9900