L3HARRIS™

WHAT'S NEW IN ENVI 5.6.1 AND IDL 8.8.1

August 2021

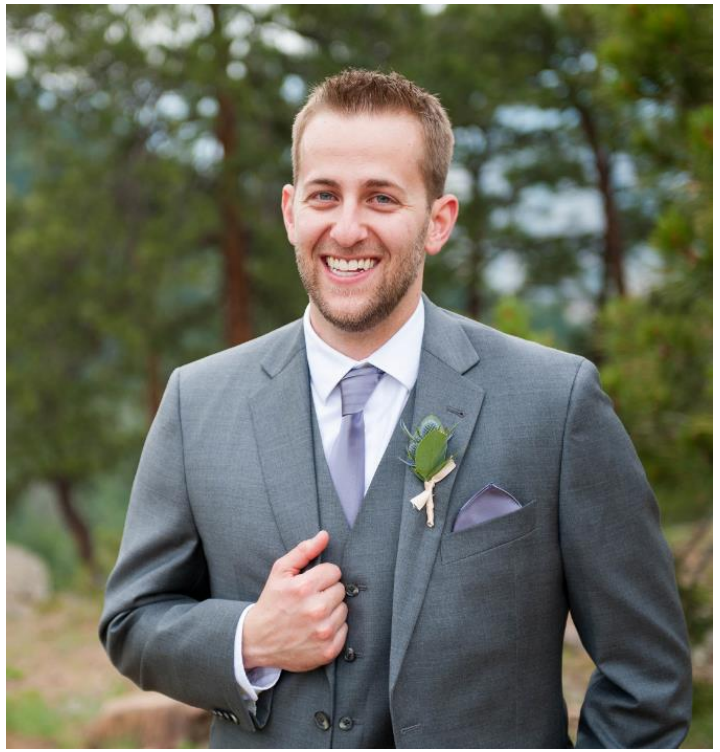
Bill Okubo | Product Manager
Zachary Norman | Product Manager

Contact Information and Introductions

Bill Okubo

Product Manager

bill.okubo@l3harris.com



Zachary Norman

Product Manager

zachary.norman@l3harris.com

Agenda

Sensor Support and New Data Formats

NITF, NITF Browser, and MIE4NITF

Time Series Tutorial

Data Connectivity

IDL: List, Hash, Orderedhash, Dictionary

ENVI Performance Improvements

Notables and Platform Support

Windows Installation Updates

Questions and Discussion

Support for Maxar Analysis-Ready Data (ARD)

ENVI 5.6.1 adds support for Maxar's ARD data

On the right you will see an animation of an ARD dataset opened in ENVI

Improved Support for Landsat Data

Supported Landsat Data Types

Data Type	Menu Option	File to Select
Landsat 1-8 Collection 2, Level 1, GeoTIFF with metadata	File > Open File > Open As > Optical > Landsat > GeoTIFF with Metadata	*_MTL.txt
Landsat 4-8 Collection 2, Level 2, GeoTIFF with metadata	File > Open	*_MTL.txt
Landsat 4-8 Collection 1 Analysis Ready Data (ARD): • Surface Reflectance • Top of Atmosphere (TOA) Reflectance • TOA Angles • TOA Brightness Temperature • Provisional Surface Temperature • Quality Assessment	File > Open	*.xml
Landsat 4-8 Collection 1, Level 3 products: • Burned Area • Dynamic Water Surface Extent • Fractional Snow Covered Area	File > Open	*.xml

Other Sensor Support

PlanetScope PS2.SD and PSB.SD

Pleiades NEO

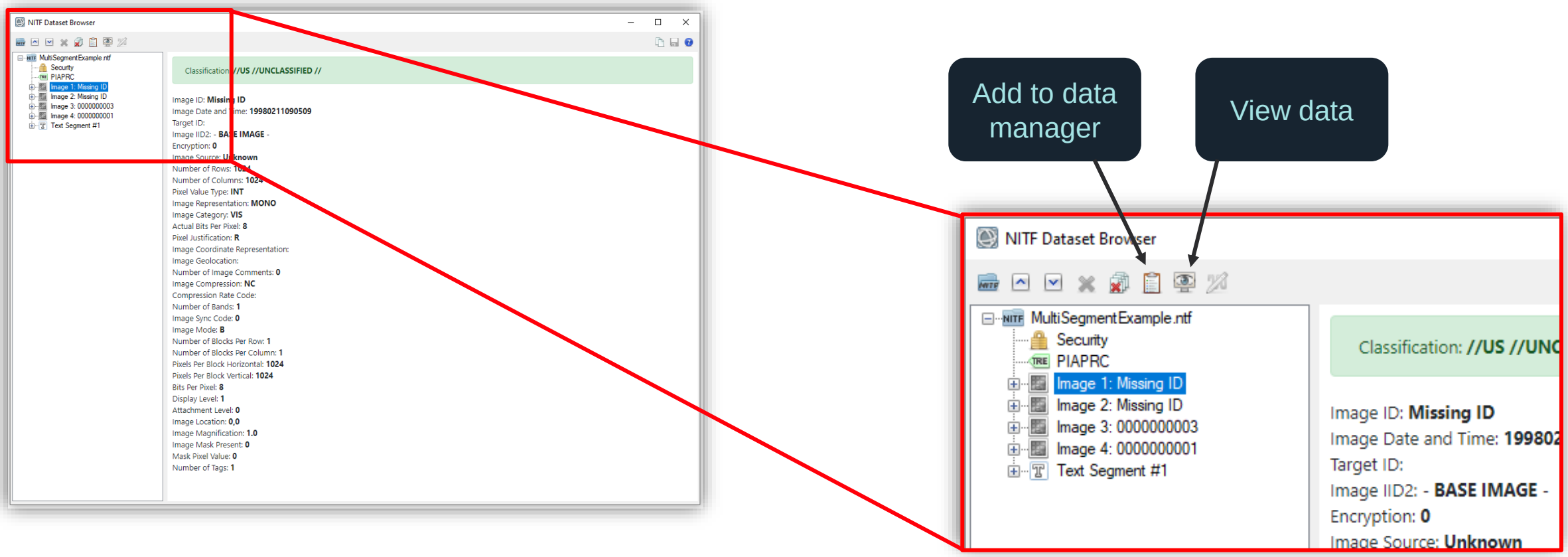
NITF in ENVI: Performance and User Experience

The new NITF experience in ENVI:

1. **Quickly open NITF and MIE4NITF files with the NITF browser**
2. **View metadata and/or open datasets**
3. **Quickly visualize and process the data**

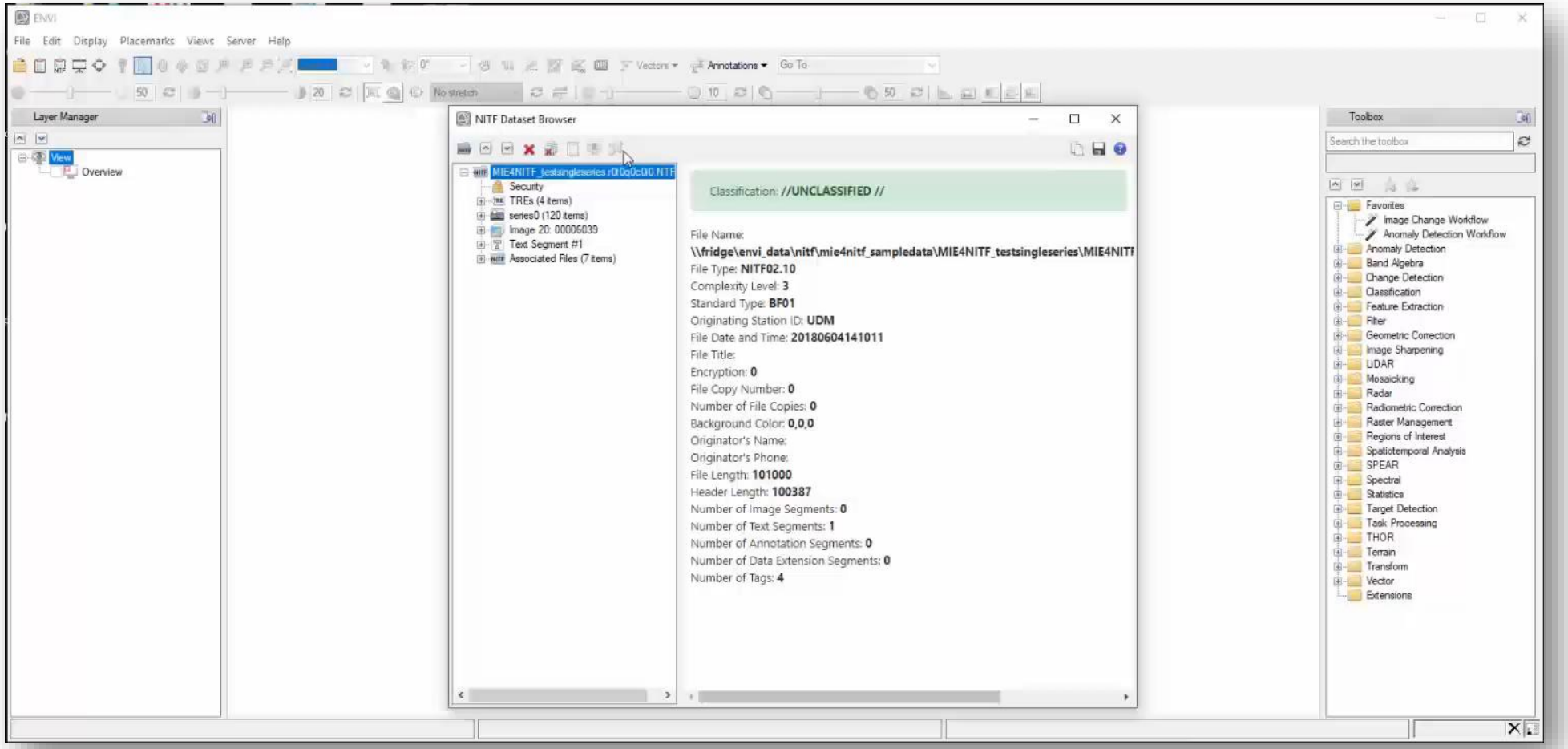
NITF type	Open	Initial Display
J2K PAN	80% faster	Display in 10 sec or less
J2K MSI	79% faster	10 + seconds
HSI Uncompressed BSQ	80% faster	Display in 10 sec or less
HSI Uncompressed BIP	83% faster	75% faster
HSI Uncompressed BIL	82% faster	70% faster
MSS - BIL	85% faster	77% faster
MIE4NITF (125 frames)	82% faster	Display in 10 sec or less
HSI Uncompressed Big Block NITF (BIP)	90% faster	20% faster
HSI Uncompressed Big Block NITF (BIL)	90% faster	20% faster
SIDD Uncompressed PAN Big Block NITF	93% faster	Display in 10 sec or less

NITF Browser



Use the NITF Browser to select frames from a MIE4NITF file, annotate it, and export a video of your AOI to PowerPoint

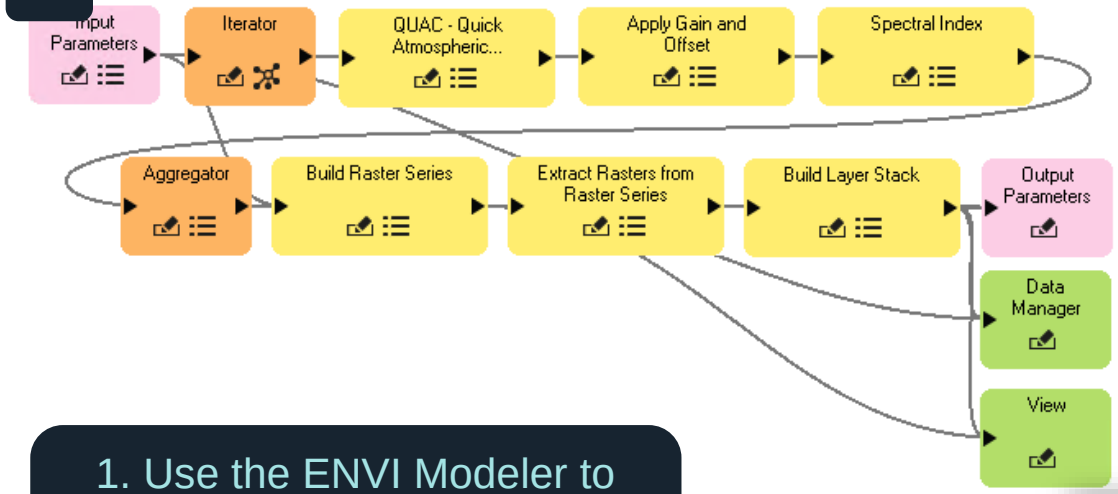
NITF Browser and PowerPoint Generation



Use the NITF Browser to select frames from a MIE4NITF file, annotate it, and export a video of your AOI to PowerPoint

Time Series Tutorial

1

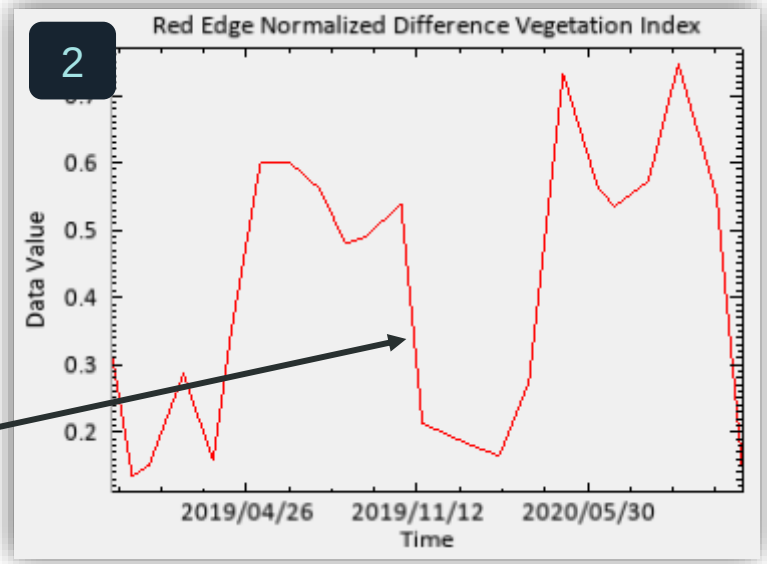


1. Use the ENVI Modeler to customize your time-series workflows

2. Using raster series in ENVI lets you look at pixel values over time

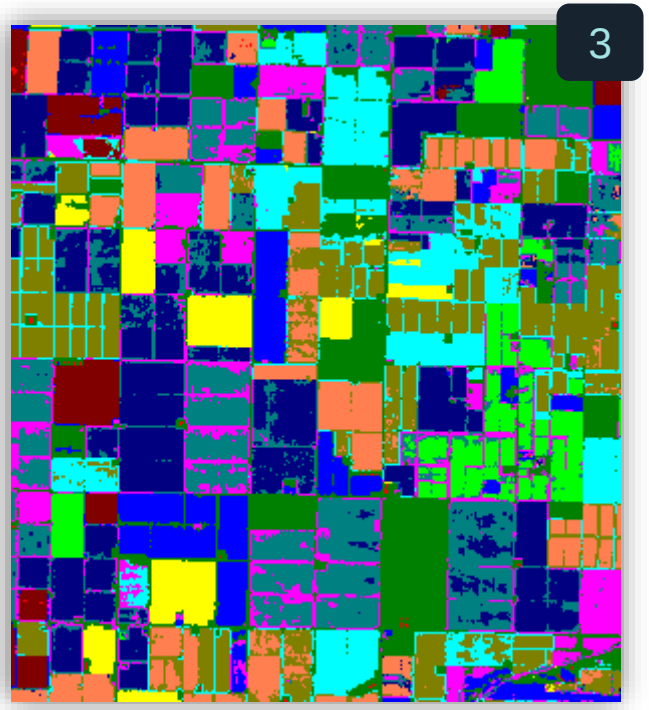
3. Find fields with similar temporal signatures using unsupervised classification

2

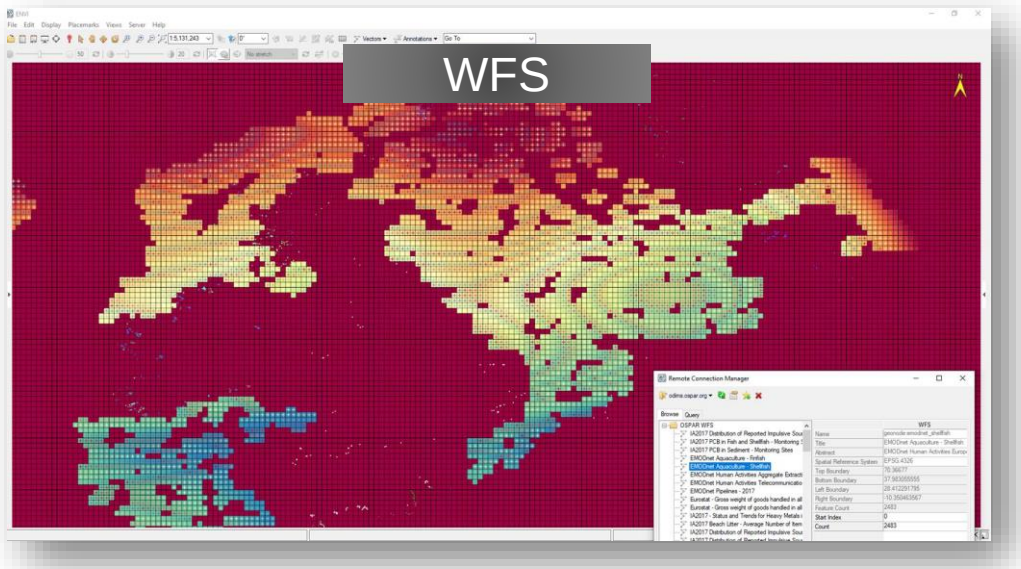


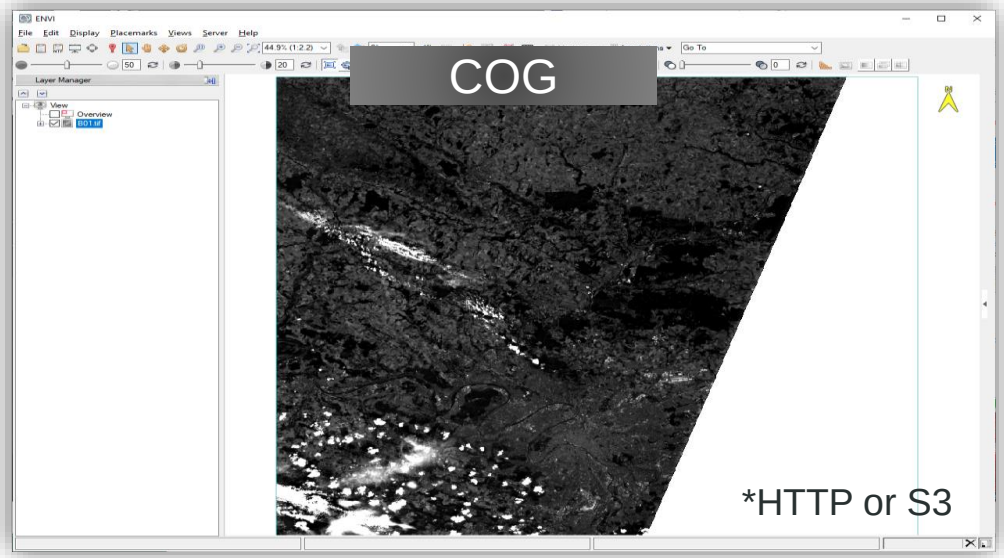
Likely when field was harvested

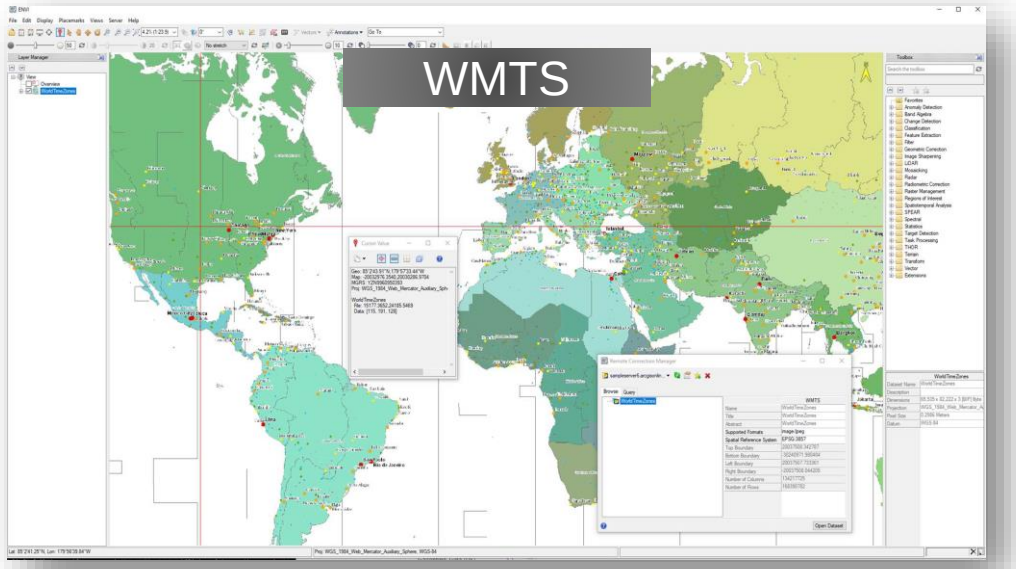
3

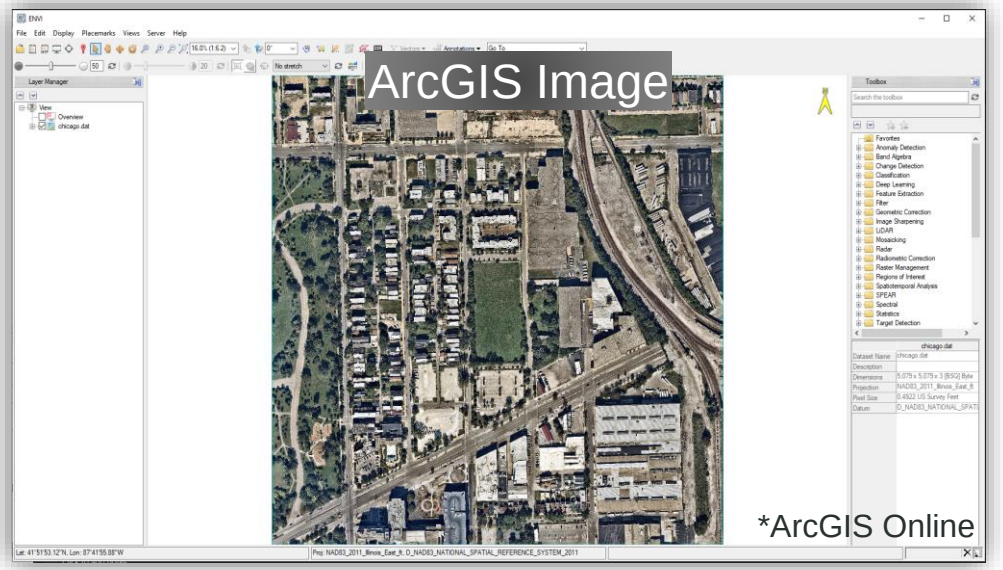


Data Connectivity









IDL: Updated Modern Data Structures

- List, hash, orderedhash, and dictionary have all been moved into core IDL!
- This brings dramatic performance improvements
- Originally, they were written in PRO code
- We did this because performance was sub-par compared to other programming languages

```
; list
l = list()

; hash and orderedhash
h = hash()
h = orderedhash()

; dictionary
d = dictionary()
```

List	8.7.3 (s)	8.8.1 (s)	% Faster
Creation	15.5	2.2	86%
Get random elements	11.0	9.3	15%
Set sequential elements	9.5	6.8	29%
Set random elements	12.5	7.0	44%
Add scalar	14.9	12.0	19%
Add array	11.7	10.4	11%
Add array with extract	9.2	4.1	56%
Count	10.6	2.7	74%
IsEmpty	9.1	1.8	81%
Move	10.1	0.1	99%
NestedMap	9.2	3.7	60%
Reduce	16.2	7.4	54%
Reduce with cumulative	7.7	3.2	58%
Remove	10.1	7.7	24%
Reverse	15.6	0.8	95%
Sort	13.8	12.4	10%
Swap	10.1	0.7	93%
ToArray	12.8	1.2	91%

Hash and Orderedhash Performance

Hash	8.7.3 (s)	8.8.1 (s)	% Faster
Creation	105.8	9.1	91%
Add element	94.5	16.4	83%
Get element	35.8	9.4	74%
Count	25.8	13.1	49%
Filter	118.9	13.3	89%
HasKey	40.9	15.2	63%
IsEmpty	27.9	15.5	45%
IsFoldCase	27.8	16.6	40%
Keys	62.4	13.2	79%
Map	161.2	17.4	89%
Reduce	130.9	14.3	89%
Remove	106.7	15.2	86%
ToStruct	47.3	13.5	71%
Values	61.7	11.3	82%
Where	66.6	14.3	79%

```
; dictionary
d = hash() ; or orderedhash

; set data
d['key'] = 42

; get data
foo = d['key']

; get all keys and values
keys = d.keys()
values = d.values()
```

Orderedhash	8.7.3 (s)	8.8.1 (s)	% Faster
Creation	91.2	11.7	87%
Add element	150.9	19.7	87%
Get element	42.7	9.6	77%
Count	26.3	12.8	52%
Filter	125.9	15.1	88%
HasKey	41.3	16.8	59%
IsEmpty	28.7	17.8	38%
IsFoldCase	28.6	16.4	43%
Keys	299.6	11.7	96%
Map	175.1	16.7	90%
Reduce	78.3	14.1	82%
Remove	161.2	14.7	91%
ToStruct	113.9	12.5	89%
Values	299.8	11.3	96%
Where	127.7	13.4	90%

Dictionary Performance

```
; dictionary
d = dictionary()

; set data
d.key = 42
d['otherKey'] = 42

; get data
foo = d.key
bar = d['key']

; get all keys and values
keys = d.keys()
values = d.values()
```

Dictionary	8.7.3 (s)	8.8.1 (s)	% Faster
Creation	90.9	14.8	84%
Add element	109.5	29.3	73%
Get element (key)	41.2	16.6	60%
Get element (dot)	69.1	12.9	81%
Count	26.2	12.1	54%
Filter	135.0	15.4	89%
HasKey	63.1	42.7	32%
IsEmpty	28.5	15.4	46%
IsFoldCase	28.4	15.3	46%
Keys	48.1	21.2	56%
Map	184.7	18.8	90%
Reduce	143.8	14.4	90%
Remove	143.7	30.9	78%
ToStruct	43.4	15.6	64%
Values	62.0	10.7	83%

Breaking Change!

RESTORING ROUTINES FROM SAVE FILES

- You *will not* be able to restore any IDL SAVE files with list, hash, orderedhash, or dictionary routine definitions in them
- Before you can use them in IDL 8.8.1, you'll need to run the "savefile_cleanup" procedure

```
; save compiled routines
save, /ROUTINES, FILENAME = 'C:\my\file\data.sav'

; remove routine definitions for objects that are now in core IDL
savefile_cleanup, 'C:\my\file\data.sav'
```

RESTORING VARIABLES FROM SAVE FILES

- For SAVE files containing variables of list, hash, orderedhash, or dictionaries, you'll need to use the RELAXED_STRUCTURE_ASSIGNMENT keyword for them to load correctly

```
; save variables as list/hash - IDL 8.8 or before
data = list()
save, data, FILENAME = 'C:\my\file\data.sav'

; restoring variables in IDL 8.8.1, use relaxed stucture
restore, 'C:\my\file\data.sav', /RELAXED_STRUCTURE_ASSIGNMENT
```

However, you **do not need** to change any IDL code to use these changes.

It is only SAVE files that will have breaking changes.

JSON Parsing

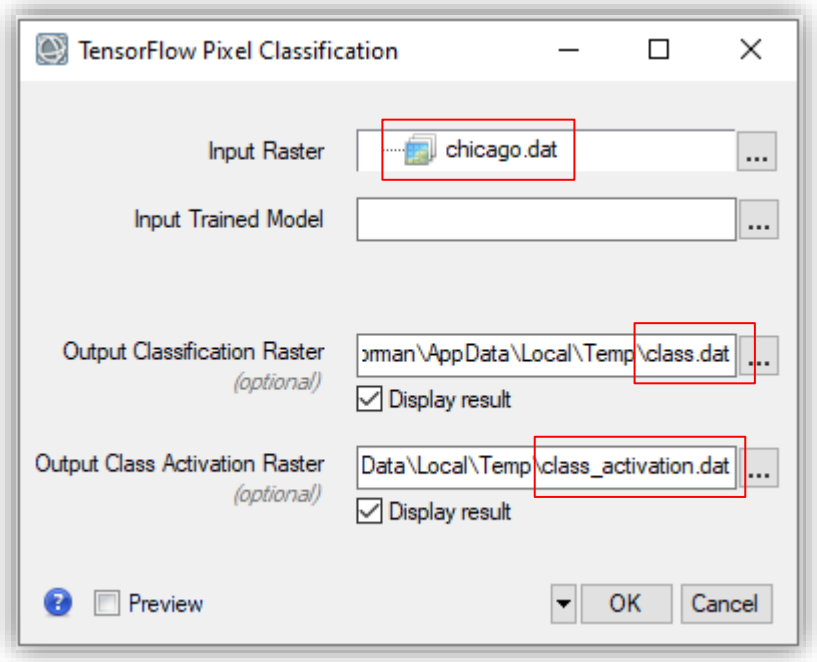
Files	String Characters (k)	IDL 8.7.3 Time (s)	IDL 8.8.1 Time (s)	% Faster	IDL 8.7.3 Chars/second	IDL 8.8.1 Chars (k)/second
82	1,221	3.22	0.42	87%	379k	2,935k
577	1,532	3.32	1.02	69%	462k	1,512k
1647	2,617	7.31	2.52	66%	358k	1,037

Parsing JSON data dramatically benefits from the work with list, hash, orderedhash, and dictionaries.

The amount of time saved depends on the size and complexity of your JSON data

ENVI Formatted Files: Quick Background

- ENVI “formatted” files are used for the outputs of most processing within ENVI
- They are binary files on disk that use interleave to structure their data for fast access depending on your application and type of data
- Common extensions: .dat, .img, or no extension with .hdr file next to it



Example: ENVI formatted files for input and output with ENVI Deep Learning

ENVI File Format Performance

Speed Test or Algorithm	BSQ Interleave			BIL Interleave			BIP Interleave		
	5.6	5.6.1	% Faster	5.6	5.6.1	% Faster	5.6	5.6.1	% Faster
Reading: Square tile (1024 x 1024)	13.9	10.7	23%	31.0	4.2	86%	100.8	3.4	97%
Reading: Single Row (3115 x 1)	20.4	6.4	69%	22.2	4.6	79%	50.6	4.5	91%
Reading: File at once	16.7	5.2	69%	39.5	3.2	92%	73.7	3.3	96%
ISOData Classification	1816.1	1725.1	5%	1937.9	1846.7	5%	2121.9	1776.1	16%
QUAC	248.6	200.4	19%	396.2	255.5	36%	473.8	269.4	43%
Spectral Angle Mapper	37.8	24.2	36%	39.1	28.0	28%	55.4	20.0	64%
Statistics with Histograms	69.0	53.8	22%	72.2	59.5	18%	97.4	95.0	2%
Mosaic w/ color balancing and feathering	2294.1	476.6	79%	2449.3	506.3	79%	2788.8	546.9	80%
Mosaic w/o color balancing and feathering	180.5	53.7	70%	188.3	72.3	62%	226.3	41.6	82%

Notables

- IDL Python Bridge Now supports Python 3.9
  - IDL 8.8.1 will support Python 3.7, 3.8, and 3.9
- ENVI startup time improvement
  - ENVI's UI launches up to 50% faster and headless ENVI launches up to 33% faster
- Spring cleaning!
  - Updated ~20 tools in the toolbox
- Support for custom plot functions in ENVI
- Annotation API and new fill options for annotations

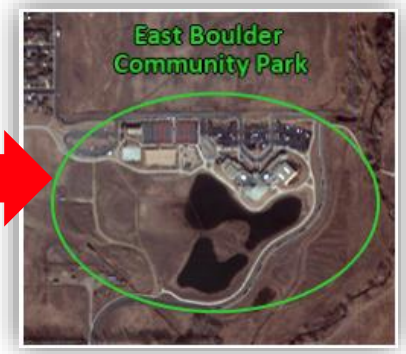
```
annotation = ENVIAnnotationSet(/GEOGRAPHIC)

annotation.AddCircle, $
  -105.22101050, 39.9917, $ ; X and Y coordinates
  [300.0, 200.0], $ ; ellipse axes (m)
  LINE_COLOR='lime green', NAME='Park polygon'

annotation.AddText, $
  -105.22118409, 39.99424191, $ ; X and Y coordinate
  'East Boulder!C Community Park', $
  GLYPH_COLOR='lime green', NAME='Park label'

outFile = e.GetTemporaryFilename()
annotation.Save, outFile

AnnLayer = View.CreateLayer(annotation)
```



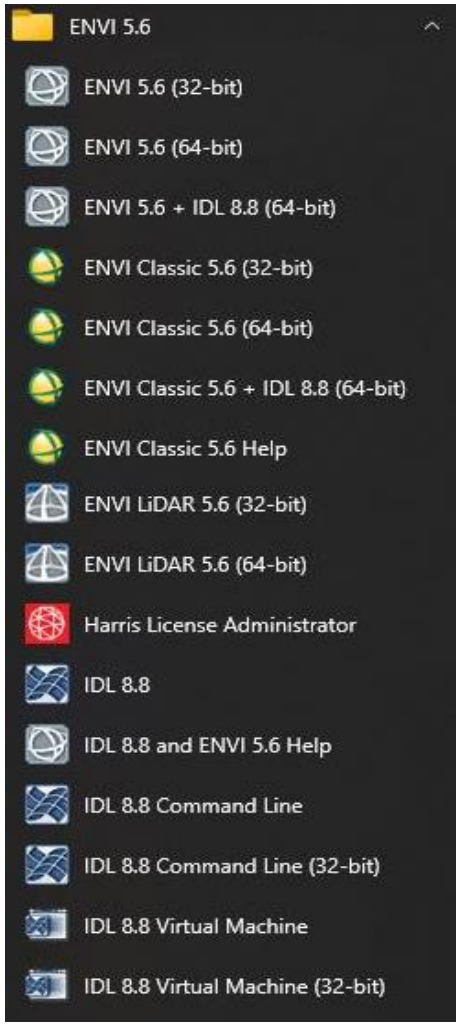
- Platform support
  - Apple M1 via Rosetta 2
    - This is ***not*** native support for ARM processors with ENVI and IDL
    - There will be some workarounds and a help article will be coming to our website
    - https://www.l3harrisgeospatial.com/Support/Self-Help-Tools/Help-Articles/Help-Articles-Detail/ArtMID/10220/ArticleID/24091/preview/true
  - Initial testing for Windows 11 shows no major issues

Simplified Start Menu on Windows

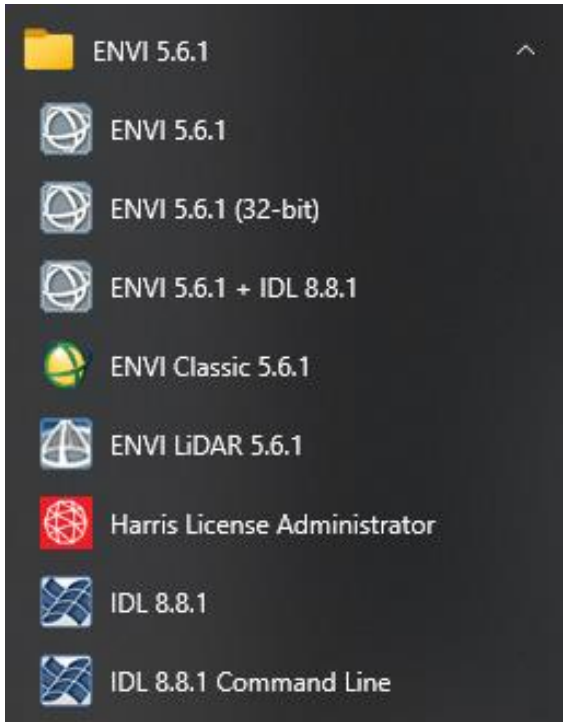
For new users, and after a change to Windows, the start menu for ENVI and IDL has been confusing without nested folders

We cleaned this up to make it less confusing!

The old shortcuts are still included in the installation folder



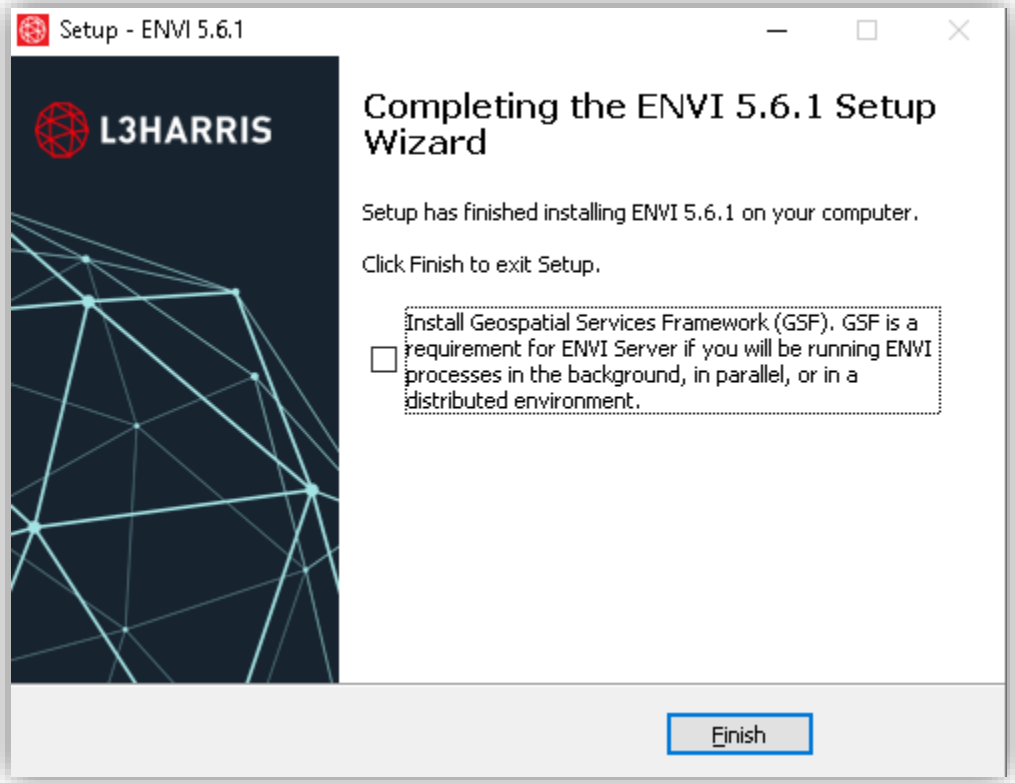
Old

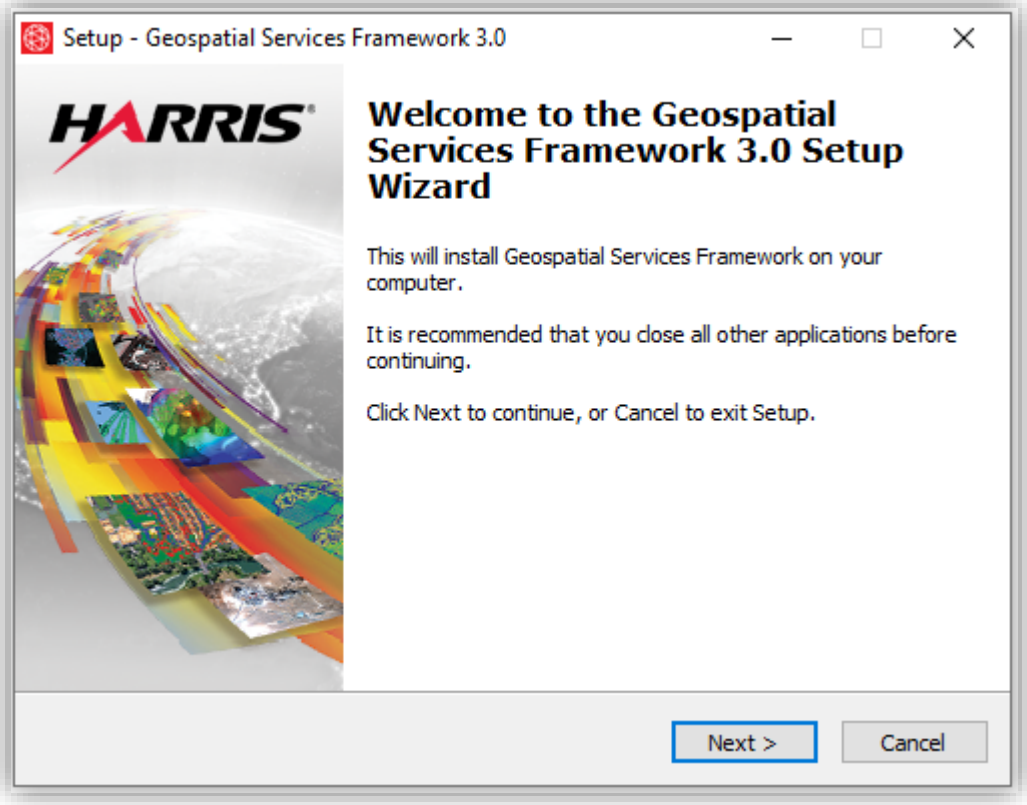


New

Updated Windows Installer for ENVI

ENVI 5.6.1 installers for Windows will come as a ZIP file with everything you need to use ENVI Server out-of-the-box





Bill Okubo

Product Manager

bill.okubo@l3harris.com

Zachary Norman

Product Manager

zachary.norman@l3harris.com

L3Harris Geospatial

www.L3HarrisGeospatial.com
geospatialinfo@l3harris.com
303-786-9900