

RELEASE NOTES

Altair[®] Compose[®] 2025

New Features and Enhancements 2025

Release Highlights

DICOM Library *

The DICOM Library reads image data and stores information from DICOM (Digital Imaging in Medicine) files.

It lets you read and modify DICOM data using Compose.

Functions include:

- **dicomread** - to read the image data from dicom file

```
1 img_path = '0015.DCM';  
2 x = dicomread(img_path);  
3 imshow(x)
```



- **dicominfo** - Read metadata from dicom file
- **dicomuid** - Generate unique Dicom id
- **dicomlookup** - lookup for specific tag/key
- **isdicom** - check if file is dicom standard
- **dicomupdate** - update attribute value in metadata struct of dicom file
- **dicomanon** - anonymize dicom file

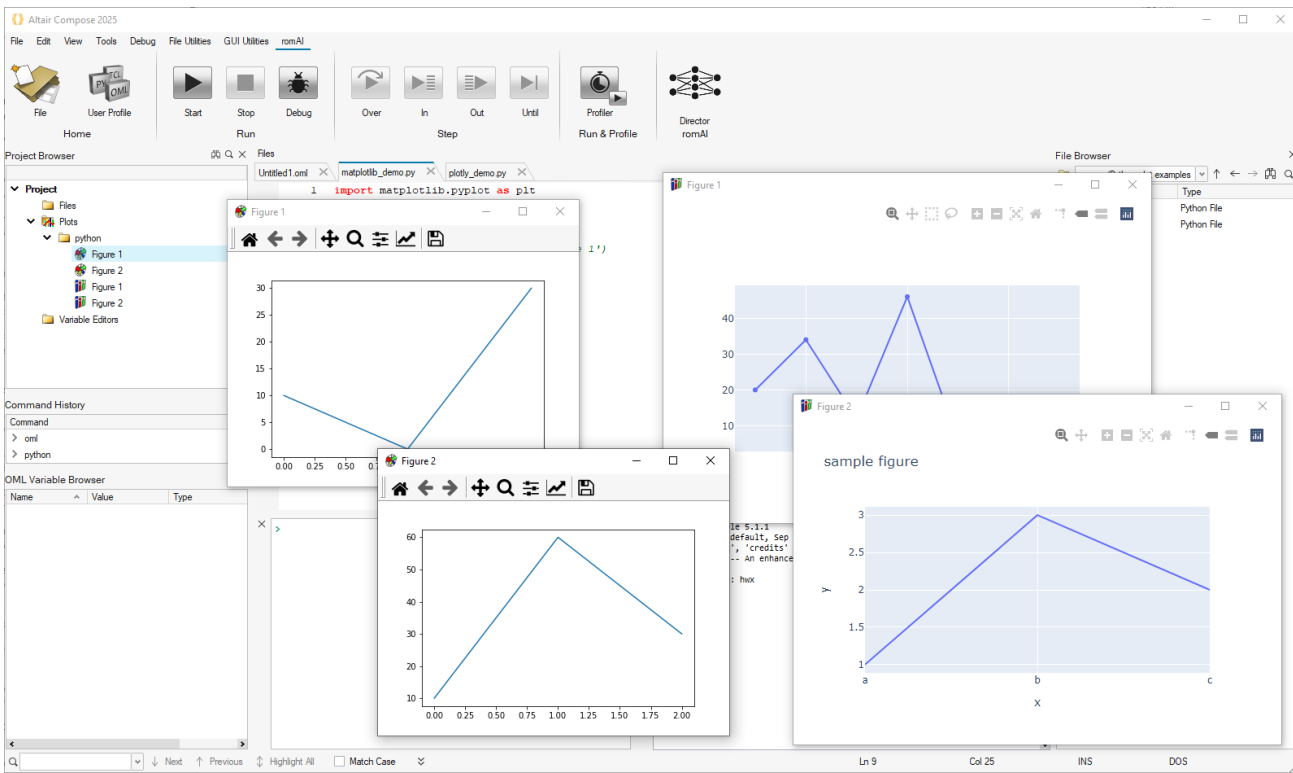
You can create a GIF from multiple DICOM files using `dicomread` and `imwrite`.

```
dcm_files = dir('*.*.dcm');  
img_path = dcm_files(i).name;  
x = dicomread(img_path);  
imshow(x);  
sz = size(getframe(gcf).cdata);  
num_frames = numel(dcm_files);  
f = zeros(sz(1), sz(2), sz(3), num_frames);  
for i = 1:numel(dcm_files)  
img_path = dcm_files(i).name;  
x = dicomread(img_path);  
imshow(x);  
f(:,:,,i) = getframe(gcf).cdata;  
end  
imwrite(f, 'gifDefault.gif');  
imwrite(f, 'gifDelay.gif', 'DelayTime', 0.5);
```

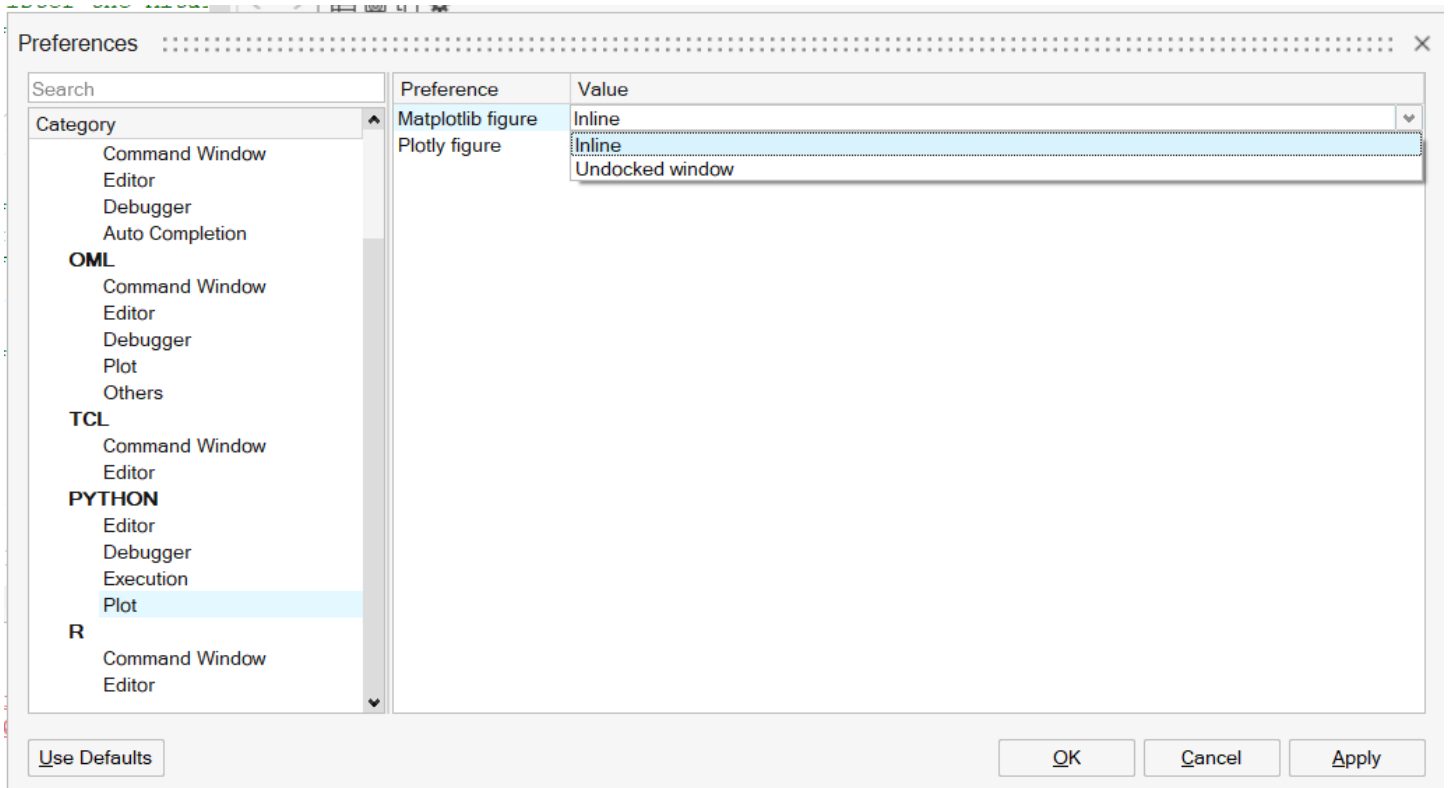
[Click to see animation.](#)

Plotting – Python

Matplotlib and Plotly plots are now supported as undocked windows.



Go to File > Preferences > Python > Plot to select how to display the python plot.



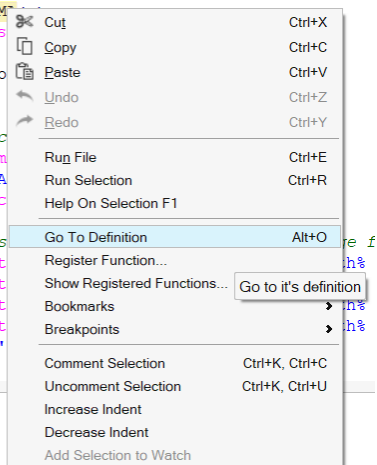
Editor

- Automatic insertion of brackets is now supported in the Compose editor for OML, Python, and TCL.
 - Closed bracket], parenthesis), and flower bracket } are automatically created.
 - In OML, a comment line starting with % is treated as a special case to not automatically insert brackets.
- Go To Definition
 - Starting with 2025, Compose IDE provides an option to right click a function and select **Go To Definition** to open the file in which the function name is defined.

```

6 %Filter function
7 function y = SMA(b, ones)
8   b = 0.2*ones;
9   a = 1;
10  y = filter(b, ones, ones);
11 end
12
13 %CAE Data extraction
14 cd(fileparts('path', 'CAEData', 'CAEData.mat'));
15 filePath = './CAEData.mat';
16 y = readmultivec(filePath);
17
18 %Extract interesting features
19 y_LT1 = SMA(y, 1);
20 y_LT2 = SMA(y, 2);
21 y_LT3 = SMA(y, 3);
22 y_LT4 = SMA(y, 4);
23 x = extract(y, y_LT1, y_LT2, y_LT3, y_LT4);
24

```



- Compose OML error info is now hyperlinked to the line in the file where the error is.

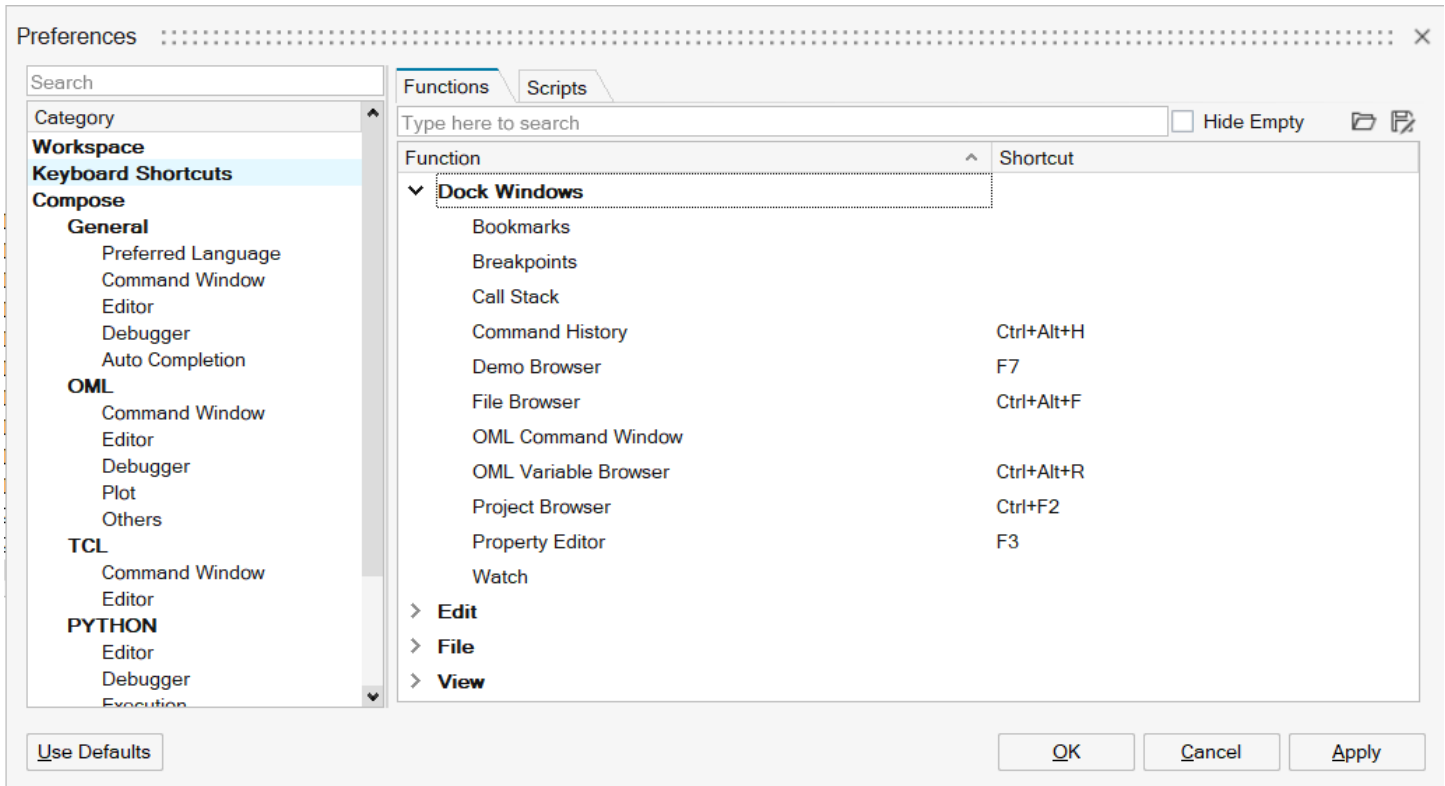
Compose Extension for AI Studio

- Support double and char as output in Compose extension to map with AI Studio datatypes.
- Support float numbers as same as number format set in AI Studio.

Keyboard Shortcuts Editor

Keyboard Shortcuts Editor to introduce new shortcuts or overwrite existing ones to run functions from ribbons and toolbars. It is also possible to run Compose OML, Python, TCL, or R scripts with a shortcut.

You can access the Shortcuts Editor from File > **Preferences**. Only Dock, File, Edit, and View section shortcuts are supported for editing.



OML

New OML Commands

Math

- Complex numbers are now supported for different math functions.
- **eigs**: Partial Eigen decomposition of a matrix.
- **maxk**: Function returns the k maximum elements of each vector in the specified dimension. If the second output is requested, it contains the index of each maximum element.

```
> m = [4,12,9,3,8,10;2,7,6,11,5,-8]
m = [Matrix] 2 x 6
4 12 9 3 8 10
2 7 6 11 5 -8
> [v,idx] = maxk(m, 5, 2)
v = [Matrix] 2 x 5
12 10 9 8 4
11 7 6 5 2
idx = [Matrix] 2 x 5
2 6 3 5 1
4 2 3 5 1
```

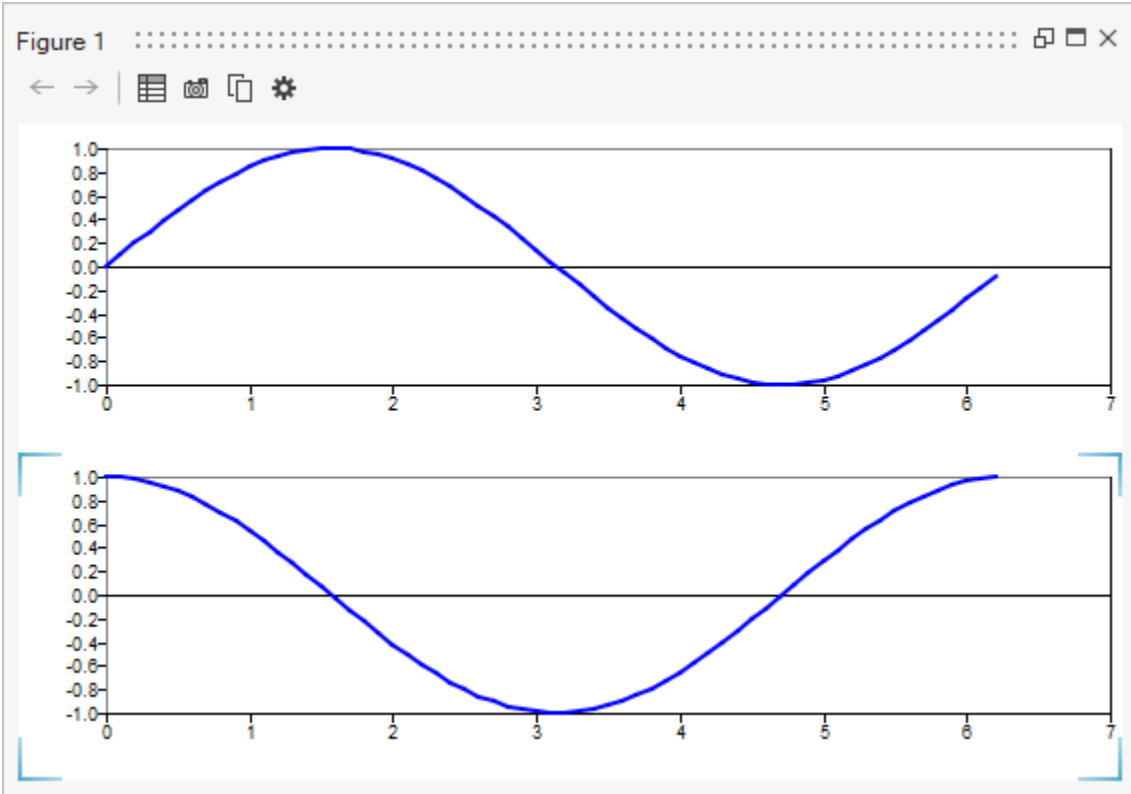
- **mink**: This function returns k minimum values from a vector or matrix.

```
> m = [4,12,9,3,8,10;2,7,6,11,5,-8]
m = [Matrix] 2 x 6
4 12 9 3 8 10
2 7 6 11 5 -8
> [v,idx] = mink(m, 5, 2)
v = [Matrix] 2 x 5
3 4 8 9 10
-8 2 5 6 7
idx = [Matrix] 2 x 5
4 1 5 3 6
6 1 5 3 2
```

- Symmetric flag in `ifft` is now supported.

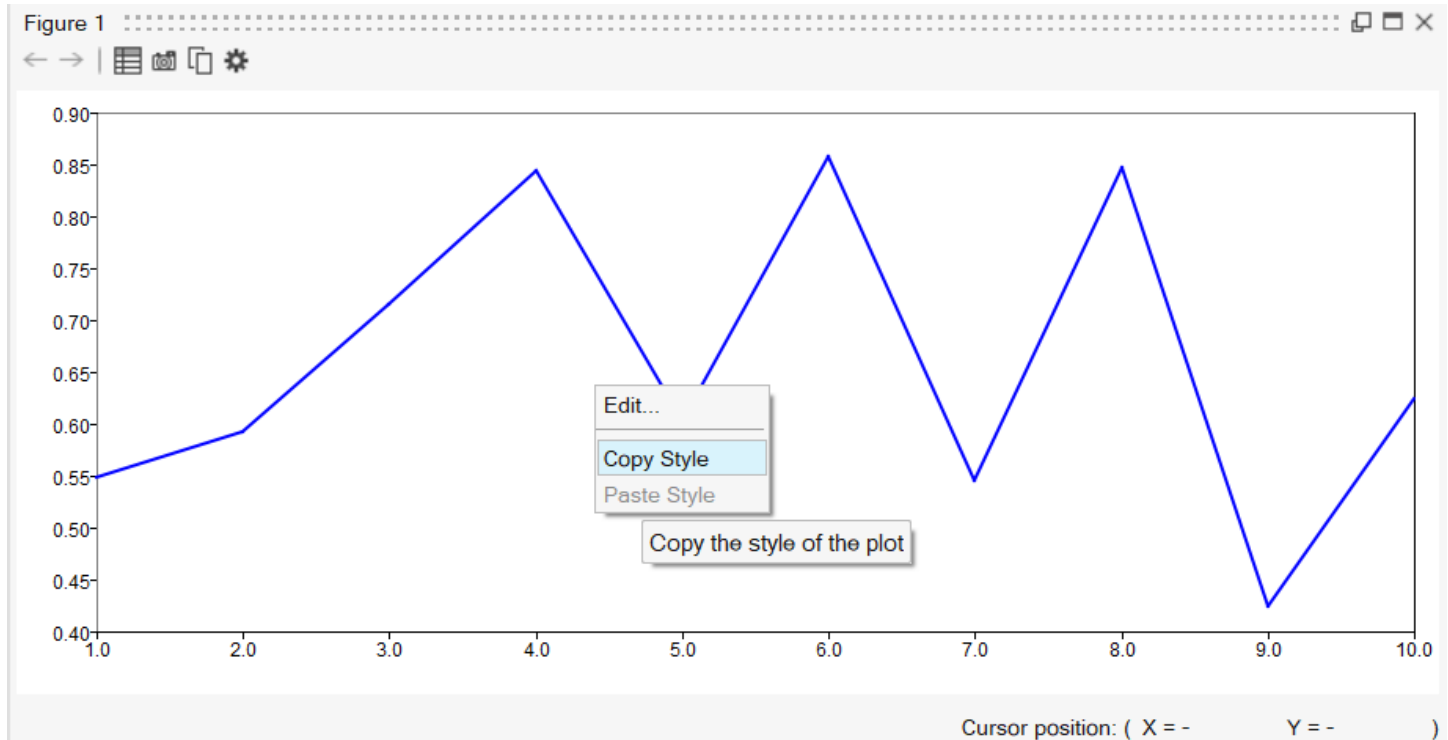
Plotting

- toolbar is now supported for OML figures, which provides commonly-used plot actions.
 - The toolbar is shown by default if the figure contains a plot, and it is hidden if a uicontrol is added.
 - To show or hide the toolbar, set the Figure property 'toolbar' to 'on' or 'off'.
`set(h,'toolbar','on')`
 - In the case of multiple subplots, the Back and Forward actions are applied to the active subplot. The active subplot is indicated by the blue lines on the subplot corners. Click on a subplot to activate it.



Icon	Action	Description
	Back	Go back to the previous view. Activated when zooming in the plot.
	Forward	Go to the next view. Activated when 'Back' button is pressed.
	Save data	Saves the plot data to a .csv or .mat file. Invokes a File browser dialog to select the file path, file name and file type to save.
	Save image	Saves the plot as an image. Invokes a File browser dialog to select the file path, file name and file type to save.
	Copy image	Copies the figure image to clipboard.
	Settings	Invokes the Figure browser dialog.

- The copystyle and pastestyle OML functions are supported from the right-click context menu for OML plots to copy and paste figure styles from one figure to another.



Enhancements

Enhancements for OML

CAE Reader 3D *

- You can now extract mid-side node results for Ansys result files using `readcae3d` by specifying `'msn_results'` in corner data.
 - If mid-side node results exist in the result file, and if you want to extract a mid-side node result, all you have to do is set `corners = "only"` and this will output mid-side node results.
 - If mid-side node results do not exist in the result file, then:
 - If you need to extract mid-side node results irrespective of averaging method selected, you set corner property value as `struct('corner', 'only', 'msn_results', 'linear')`.
 - If you do not want to extract mid-side node results for specific averaging method, you set corner property value as `struct('corner', 'only', 'msn_results', 'ignore')`.
- Performance improvements and new datatypes are now supported for `op2`, `hdf5`, and `xdb` result files in `gettypelist3d` and `readcae3d` commands. The order of datatypes could change for these result files.

Table class enhancements

- Indexing of table class variable:
 - $X(\text{rows}, \text{vars})$ = outputs table; extracts a portion of table by specifying rownames/indices and varnames/indices.
where
rows = index or string of RowName
vars = index or string of Varnames
- table variable echo in the command window is right-aligned.

```
> t
```

```
t = 5x3 table
```

```
      Lake  Area Volume
-----
'Superior' 31700  2900
'Michigan' 22300  1180
  'Huron'  23000   850
   'Erie'  9910   116
'Ontario'  7340   393
```

Resolved Issues

- interp error when extrapolating
- Temp directory not deleted after using Compose extension in AI Studio
- CLONE - Execute Compose Operator - cd in template script results in no outputs
- table variable echo in command window should Align right.
- coordinates output from readcae3d is not matching with HV result
- UITable: Column width and height is not proper
- CAE Reader 3D: Error message shows text as "rData: Resources: ... "
- Nested function call crashes
- table indexing with end operator inside parenthesis
- Can't debug python script repeatedly in the existing session w/o relaunching the Compose Debugger
- Variables shared by function and its nested functions not working correctly
- Wrong shortcuts in Shortcuts editor in Compose
- Text selection highlight color need to be different in Dark Mode for Plot Assistant and Signal Viewer command window
- Uninstall extension should remove the demo folder from demo browser
- mpower does not allow taking the power to an imaginary number

* *Applies to Standard Edition only as a paid extension*