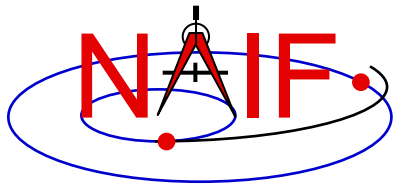


---

Navigation and Ancillary Information Facility

# **SPICE-Enhanced Cosmographia**

**April 2023**



# Cosmographia Overview - 1

Navigation and Ancillary Information Facility

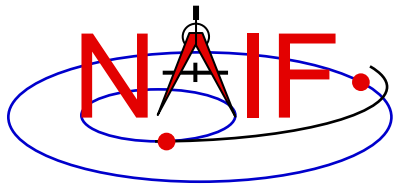
- **SPICE-Enhanced Cosmographia is an interactive 3D Solar System visualization tool**

**It can be used to render...**

- natural bodies
- spacecraft
- sensor FOVs
- observation footprints
- trajectories
- directions
- reference frames

**And has visual guides such as ...**

- stars
- longitude-latitude grids
- shadows
- reflections
- atmospheres

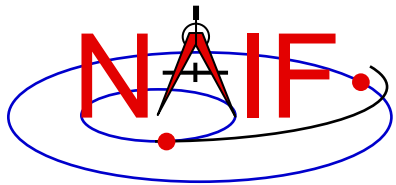


# Cosmographia Overview - 2

---

Navigation and Ancillary Information Facility

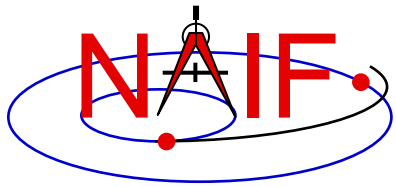
- **It is a standalone application that needs to be installed and run on a user computer**
  - **SPICE data used by the program must be available locally on that computer**
- **The program installers for Mac, Linux and Windows are available on the NAIF server**  
**<https://naif.jpl.nasa.gov/naif/cosmographia.html>**



# Cosmographia History

Navigation and Ancillary Information Facility

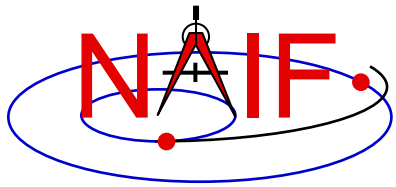
- It was originally designed as a general interest solar system simulator by Chris Laurel in 2011-2012
- In 2013-2014, with permission from the original author, JPL developers started augmenting it to use SPICE kernels to accurately visualize space mission geometry
- NAIF took over in 2014 and since then, with the help from Chris Laurel, continued its development, releasing 5 new versions to date
  - [Version 2.0, March 2015](#)
    - » Initial release by NAIF
  - [Version 3.0, December 2015](#)
    - » Added Python-based scripting, on-line User's Guide, frame and vector visualization, and more
  - [Version 4.0, February 2019](#)
    - » Upgrade to SPICE N0066, Qt5, and PyQt5; new GUI and catalog features; new scripting functions
  - [Version 4.1, February 2022](#)
    - » Upgrade to SPICE N0067; DSK support; font size controls; new scripting functions; and more
  - [Version 4.2, December 2022](#)
    - » Upgrade to Python3; addition of Python scripting support on Windows



# Cosmographia Capabilities

Navigation and Ancillary Information Facility

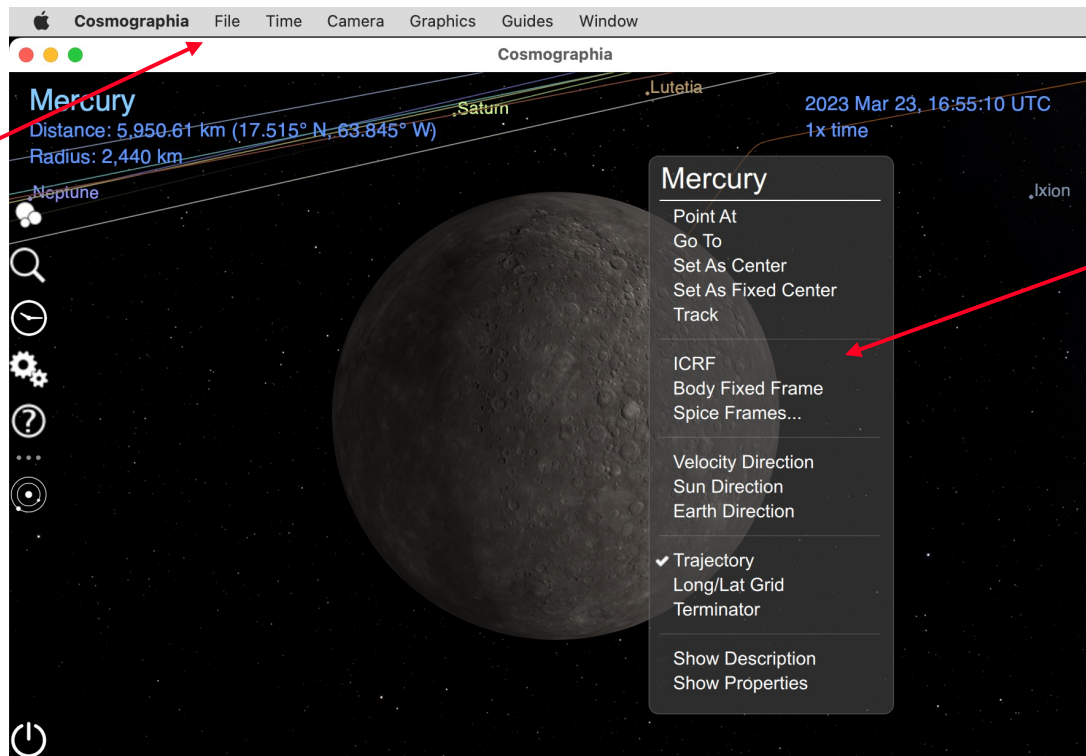
- **Uses SPICE data to ensure accurate geometry rendering**
  - Trajectories, velocities, direction vectors based on SPK
  - Orientations, reference frames based on SPICE frames
  - 3D shapes based on DSK
  - Distances, altitudes, angles based on SPICE calculations
- **Allows easy introduction of new missions by defining their SPICE data and objects in JSON-formatted catalog files**
- **Provides controls to easily manipulate time and camera position in the Solar System**
- **Includes built-in Python3 scripting interface for creating visualization sequences**
- **Has built-in screenshot and video recording (OSX only) capabilities**
- **Can be launched from a terminal/command window with a variety of options, facilitating use in automated applications**
- **And many more ...**



# Cosmographia GUI Controls

Navigation and Ancillary Information Facility

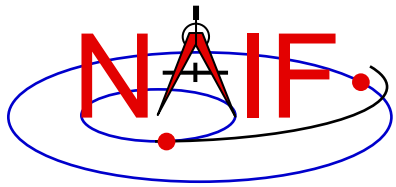
- **Cosmographia GUI provides a wide variety of controls including Top Menu, Left Toolbar, Object Context Menu, Mouse gestures, and Keyboard Shortcuts.**
  - For convenience, most Cosmographia functions can be accessed through more than one type of control



Top Menu

Left Toolbar

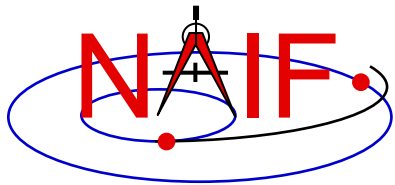
Object Context Menu (invoked with right click)



# Setting up for a New Mission

Navigation and Ancillary Information Facility

- **Setting up Cosmographia to visualize a mission based on SPICE data usually requires**
  - Downloading needed kernels to user's computer
  - Creating a JSON catalog file loading these kernels, directly or via a meta-kernel
  - Creating a JSON catalog file defining attributes of natural bodies that are the mission's targets, including attributes defining how to compute target's trajectory and orientation using SPICE
  - Creating a similar JSON catalog file for the spacecraft(s)
  - Creating JSON catalog files defining sensor parameters, if applicable
  - Creating JSON files defining sensor observation sequences, if applicable
- **With the help from Cosmographia documentation, JSON catalog templates, and example JSON catalogs this can be done with a fairly modest effort**



# Cosmographia Built-in Scripting

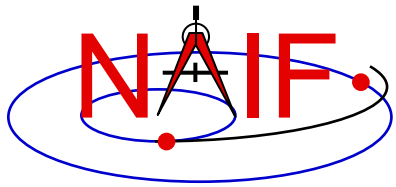
Navigation and Ancillary Information Facility

- **Cosmographia includes a Python3-based scripting interface**
  - provides access to many of the program's elementary visualization functions
  - allows users to create visualization sequences using these functions
- **This example script loads VCO SPICE data, spacecraft and LIR instrument catalog files, sets a specific UTC time, aligns program rendering view with the instrument FOV, takes a screenshot simulating an image, and exits the program**

```
import cosmoscripting
cosmo = cosmoscripting.Cosmo()
cosmo.loadCatalogFile("/Users/naif/VCO/spice_VCO.json")
cosmo.loadCatalogFile("/Users/naif/VCO/spacecraft_VCO.json")
cosmo.loadCatalogFile("/Users/naif/VCO/sensor_VCO_LIR_PIC-VENUS.json")
cosmo.setTime("2015-12-07 05:26:01 UTC")
cosmo.setFov(12.5,2.0)
cosmo.moveToPov("VCO_LIR_PIC", [0,0,0.01], [0,0,1], [1,0,0],1).wait(2)
cosmo.saveScreenShotToFile("Users/naif/VCO/cosmo_screenshot.png").wait(1)
cosmo.quit()
```

- **Cosmoscripting-based scripts can be run from GUI via menu or from command line using the “-p” option.**





# Cosmographia Documentation

Navigation and Ancillary Information Facility

- Detailed information about all Cosmographia features is available in its on-line User's Guide

<https://cosmoguide.org>



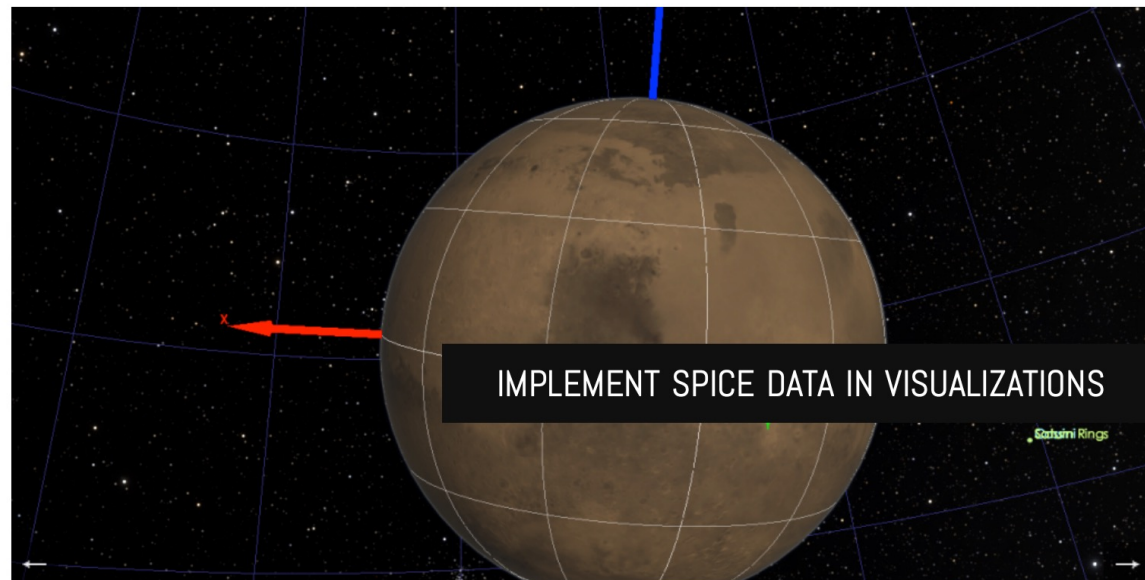
SPICE-ENHANCED

Cosmographia User's Guide

PUBLISHED ON DECEMBER 21, 2015

Search ...

- HOME
- INTRODUCTION
- INTERFACE CONTROLS
- SPICE DATA
- CREATING CATALOG FILES
- SCRIPTING
- TIPS
- GLOSSARY
- VIDEOS
- CREDITS



PAGES MOST VIEWED