

# WWT for Researchers

## High Level Technology Overview of WorldWide Telescope

- Imagery and data hosted in the cloud
  - WWT object and survey imagery can be processed by existing data curators or hosted in the cloud as a tiled multiresolution format (TOAST) to enable very fast browsing and zooming into very high resolution surveys.
  - The image tiles are accessible via WWT clients: (Windows client) or modern browsers (Win/Mac) and via API for web services e.g.: ADS All Sky Survey
  - Global content distribution network allows for fast data delivery
- High-performance Windows application (download required)
  - Guided Tour authoring
  - Supports 3D imagery
  - High performance interactive data visualization
  - Support for planetarium and virtual reality experiences
- Web Browser Client featuring: (no download required)
  - Many of capabilities of the WWT Windows application in a browser.
  - Comparable survey and object image rendering in the sky
  - Playback of 2D tours imagery
  - Limited interactive data visualization
  - Toolkit, which can be used to build custom, interactive web controls e.g.: Interactive Planck Data Explorer.

## Resources

### Getting Data into WWT

- Adding Astronomical Image Data  
<http://worldwidetelescope.org/Learn/Exploring#AstroImageData>
- Using Virtual Observatory (VO) Tables  
<http://worldwidetelescope.org/Learn/Exploring#UsingVOTables>
- Loading Data Using pyWWT <http://worldwidetelescope.org/Learn/Exploring#pyWWT>
- Tutorials linking WWT and iPython notebooks  
[https://github.com/marksubbarao/pyWWT\\_AAS225](https://github.com/marksubbarao/pyWWT_AAS225)

### Making Tours

- Getting Started with Authoring Slide-based Tours  
<http://worldwidetelescope.org/Learn/Authoring#slidebasedtours>
- Using the Timeline Editor  
<http://worldwidetelescope.org/Learn/Authoring#timelineeditor>
- Dealing with Audio <http://worldwidetelescope.org/Learn/Authoring#dealingwithaudio>
- Editing Audio <http://worldwidetelescope.org/Learn/Authoring#editingaudio>

## Leveraging and Extending WorldWide Telescope

The opening of the WorldWide Telescope codebase to open source provides several new ways of leveraging and extending WWT for your purposes.

WWT was already fairly open to data and reuse with the various software development kits, and the ability to integrate data into WWT thru the user interface and WTML (Our XML based description language to add data into WWT). The following capabilities are already available in WWT and do not require any changes to the existing source code.

Below are ways to extend WWT and the skills needed that don't require changing Open Source components using existing WWT interfaces extend its capabilities.

### Create a tour

#### What you can do

You can create a variety of linear and interactive tours with WWT. The tour authoring interface allows you to guide users thru the universe by positioning a virtual camera in various slides, and WWT animates the transitions for you. You can also add your own images, data, text, music, voice over and other media to enhance the message. Buttons, images and other elements can link to other sets of slides, allowing you to create content that allows the user to navigate. This can be used to create Kiosks, menu driven multimedia content, training and quizzing interactives and self-service data exploration. Linear tours can also be outputted to high quality video frames for professional quality video production at any resolution desired. Tours can also be hosted in a website to create interactive web content

#### Skills Required

WWT tours are one of the most powerful aspects of WWT and don't require any programming skills. You should know what story you want to tell and understand presentation and layout skills. If you can make a PowerPoint presentation then you should be able to make a WWT tour.

### Visualize data in WWT Layers

#### What you can do

You can bring your own data into WWT and position it anywhere in the universe using our Layer Manager. Data can include imagery placed on the ground or sky, both 2d and 3d geospatial point data on Earth, other planets, the sky and anywhere in the universe. You can use formats such as OGC WKT (Well Known Text) to draw lines, polygons and create shapes with 3d volume. You can display KML layers, as well as mapping tabular data to different visualizations. Any data can feature a temporal dimension that can be animated over time. You can import data from flat files, thru our Excel plug-in or thru KML, shape files and odata sources. You can also define and import orbits around the sun, Earth, moons, planets or even Lagrangian points. If you want to visualize spacecraft, buildings, ships or any other 3d object you can import 3d models from .obj and .3ds file formats.

#### Skills Required

You need to be familiar with your data and understand the basics of working with coordinate systems like RA/DEC, Lat/Long or X,Y,Z. You may need to understand how to massage data in a tool like Excel or thru code or scripting languages. You may need skill in doing database queries or using internet sites to extract data files. You should know enough about the data you are visualizing to confirm that unit

mappings, and coordinate mappings and scaling are correct. The Excel plugin for WWT can help make the task of importing data into WWT easier (included with the WWT application download).

## Using the LCAPI (Layer Control API) to feed data into WWT Layers

### What it can do

In addition to using the layer manager interactively, you can use scripting and programming languages to send commands and data into WWT from your own code driving the visualization of data in WWT. You can write code that can read data from a database, or create it mathematically and send it to WWT to render. In addition almost all of WWT functionality can be controlled thru the LCAPI. This means you could create control mechanism for automated or interactive control of WWT thru your own hardware and software.

### Skills Required

You need to be familiar with the programming language of your choice and get the LCAPI documentation and learn how to use HTTP request to send and receive data from WWT. There is also a library called NARWOL that provides a .NET wrapper and high level programming features for those who use .NET. Any language that can call HTTP web services can control WWT

## Using MIDI and Xbox gamepad to control WWT

### What it can do

MIDI (Musical Instrument Digital Interface) is a protocol for connection performance controllers and media generator/controllers. MIDI keyboards, control panels, foot pedals or any other MIDI capable device can be connected to a MIDI or USB input where WWT is installed and those can be used to control WWT functionality. WWT provides a setup panel for you to map keys, buttons, knobs, sliders and foot pedals to nearly all of WWT functionality. Similarly, WWT allows custom mapping to an Xbox game controller. This could be used in planetarium control boards, for kiosk control in a museum, for allowing researchers to explore their data with temporal and other controls mapped to relatively inexpensive controller sliders, jog/shuttle etc.

### Skills Required

You need to be able to understand how to connect your midi device to your computer, and use the WWT device configuration panel to map your Xbox controller or MIDI device to WWT functions. Note, more custom MIDI devices may require coding or hardware assembly, but off the shelf MIDI controller are plug and play with WWT.

## Open Source Development of WorldWide Telescope

WorldWide Telescope is now an open source project! The source code is now in GitHub with <https://github.com/WorldWideTelescope> as the main repository. Within the main repository, we have published a comprehensive set of code bases (see below). You can now build most of them, including the windows client from source code by simply cloning into visual studio and hitting the F5 key!

- <https://github.com/WorldWideTelescope/wwt-windows-client> - WorldWide Telescope Windows Application
- <https://github.com/WorldWideTelescope/wwt-web-client> - WorldWide Telescope Web Client lets you explore the universe in your browser!
- <https://github.com/WorldWideTelescope/wwt-website> - Website for the OpenWWT Project

- <https://github.com/WorldWideTelescope/wwt-wwtremote> - Cluster Controller for WorldWide Telescope running in multi-channel domes, power walls and other multiple machine or headless operation
- <https://github.com/WorldWideTelescope/wwt-narwhal> - Source code for Narwhal, a library and demo apps for visualization of complex datasets in WorldWide Telescope.
- <https://github.com/WorldWideTelescope/wwt-wiki> - Repository of Open WorldWide Telescope Project Information
- <https://github.com/WorldWideTelescope/wwt-tile-sdk> - SDK for large scale processing of data for WWT

To get started:

1. Get a Windows machine with about 20 GB of space.
2. Install this version (including the Setup/Deployment addin) of Visual Studio Community Edition (<https://visualstudiogallery.msdn.microsoft.com/9abe329c-9bba-44a1-be59-0fbf6151054d>).
3. Get an account at GitHub.
4. Download the GitHub client.
5. Choose the repository from the list above. Clone it to your local computer.
6. Startup Visual Studio.
7. Open the solution for the Windows client **"FILE/Open/Project-Solution"** and navigate to the location of the solutions file: e.g., C:\Users\Henry\Documents\GitHub\wwt-windows-client\WWTEplorer.sln.
8. Then build the client: **"BUILD/Build Solution."**