

WorldWind Earth v0.6.2

Operations Manual

19 January 2017

WxOps Inc.

28 Schenck Parkway, Suite 200
Asheville, NC 28803-5088

Rights Conveyed

You are granted rights to use and distribute the WorldWind Earth github content in accordance with the NASA WorldWind License at <https://worldwind.arc.nasa.gov/LICENSE.html>. The WorldWind Earth github content is comprised of the WorldWind JAVA Client and extensions developed by WxOps and the WWE User Community, up to and including the WxOps COM API Emulation of the Keyhole EarthLib functions. Rights are not NOT CONVEYED for WxOps® and WxAnalyst® Proprietary materials and intellectual property, including OpsTablet®, WxAzygy®, COLLADA Computing™, US Patent No. 8,392,853, US Patent Pub. No. 20160293021, and US Patent App. No. 15/336,411. The US Government retains specific SBIR rights in US Patent No. 8,392,853 under contract NNX 09CF28P, and in US Patent Pub. No. 20160293021 under contract NNX15CA90P.

This document is approved for unlimited distribution.

WorldWind Earth v0.6.2

WxOps has released WorldWind Earth version 0.6.x to github. This release of software and test data to open source meets and exceeds the goals of WxOps' SBIR Phase 1 objectives towards an operational geobrowser-based Common Operating Environment (COE). Our SBIR innovation adds an EarthLib COM API emulation to NASA WorldWind JAVA, allowing cross-process coordination between WxOps WorldWind Earth (WWE) and third party applications. When coupled with WxOps proprietary OpsTablet®, a geobrowser-based COE enables pilots and dispatchers to coordinate flight planning changes during long-haul flights, reducing multiple risks to commercial aircraft operations. Previously available only on Microsoft Windows, WxOps EarthLib extends the WWE geobrowser-based COE to other operating systems including Linux and Mac OSX. Since WWE is a client side application and can operate offline, OpsTablet® users are provided with uninterrupted services when internet connectivity is unavailable.

WWE v0.6.2 emulates key EarthLib COM API functions for geobrowser NAVIGATION, and introduces new JAVA classes for geobrowser ANIMATION. Florida Institute of Technology students Peter Subacz (Ocean Engineering) and Vanessa Haley (Atmospheric Sciences) joined WxOps to develop WWE in JAVA under the guidance of WxOps CIO Dr. Scott T. Shipley. Of the thirty (30) primary functions released by KeyHole in EarthLib COM API, WxOps EarthLib targets thirteen (13) of these for OpsTablet® requirements, and two (2) of these are supported.

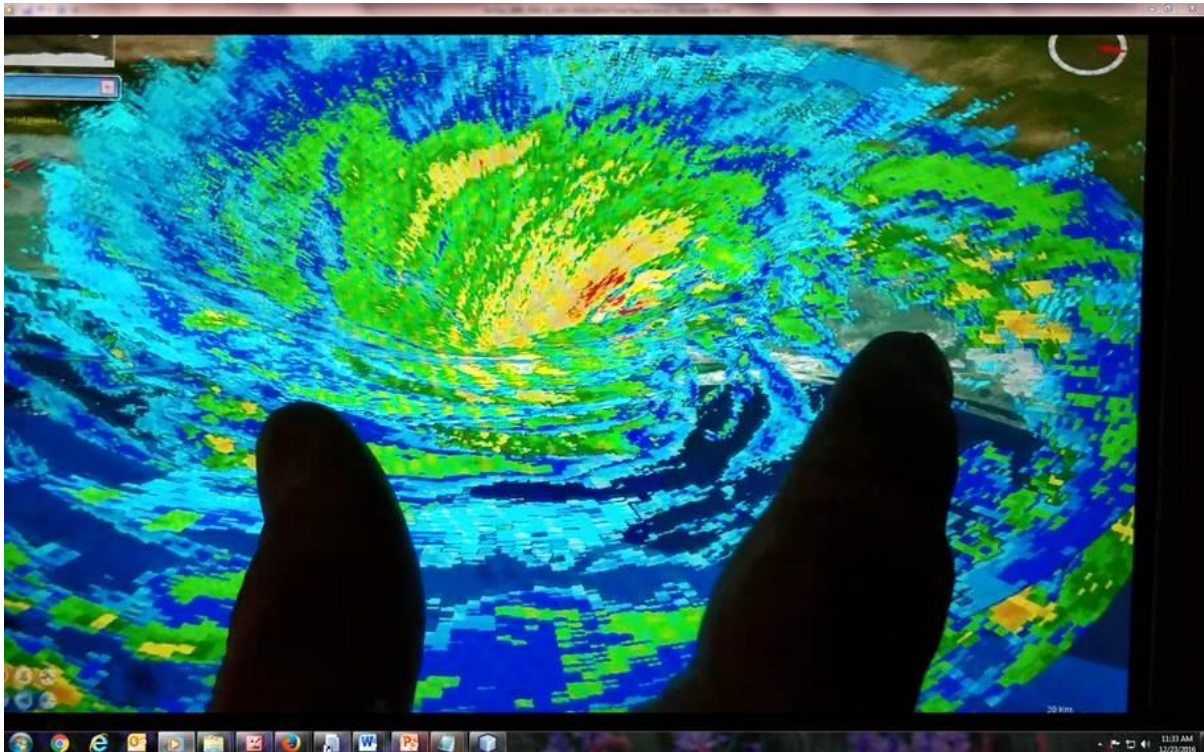


Figure 1 – WxOps WorldWind Earth (WWE) is presented by Mark Spence to the International Electronic Flight Bag (EFB) Conference in Honolulu on 3 November 2016. In this demonstration, touch navigation control is performed during the animation of NEXRAD 4DKMZ COLLADA-based KML for Hurricane Dolly (2008).

1. Introduction

This WorldWind Earth application is built using NASA WorldWind with support from WxOps Inc. and NASA under SBIR Phase 1 contract NNX16CA30P. This WorldWind Earth application visualizes weather data and performs animation by extending the NASA WorldWind JAVA Client. The main functions of this application allow users to

1. load KML & KMZ
2. manipulate the WorldWind Camera or Point of View (POV), and
3. animate KML with standard TimeSpans.

The following hyperlink is an example of animation and weather data:

<https://www.youtube.com/watch?v=od7pc1fOsoo>

WWE is maintained in JAVA as open source on github. WxOps recommends the NETBEANS IDE, or the latest JAVA Runtime JRE 8.x, as described in section 2. A description of the basic functions in WWE is provided in section 3.

2. Installation & Getting Started

How to import the project into NETBEANS 8.2 IDE and run the main class:

1. Download the project files on github.com:
<https://github.com/NASAWorldWindResearch/WxOps-WorldWind>
2. Add the project to NETBEANS IDE by selecting: File>Import Project>From ZIP
3. Once the project is imported, compile the code by pressing the "Run Project" button
4. In the Projects Tab, navigate to the main classes by selecting:
WxOpsWorldwindEarth>Source Packages> WxOps.Worldwind> WorldWindUI.java

How to Run WxOpsWorldwindEarth.jar from the command line:

1. Download the project files on github.com and compile in NETBEANS IDE
2. Open a windows command prompt.
3. Navigate to the WxopsWorldwindEarth directory and then go to the dist subfolder
4. Type the following in the command prompt: java -jar "WxOpsWorldwindEarth.jar"

Alternate approach using JRE 8:

1. Download and install JAVA Runtime (JRE) 8 from
<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>
2. Download dist.zip from <https://wxops.com/demo/dist.zip>
3. Extract dist.zip into any local folder
4. Open RUNME.BAT, or Run WxOpsWorldwindEarth.jar from the command line

3. How to Use

Start the program using one of the methods provided in section 2. The active UI for WorldWind comes with five main onscreen elements, and a bottom panel that reports information to the user. These elements are as follows:

- World Map - The World Map element is in the top left corner of the screen that reports the location of the center of the screen.
- Compass - The Compass element is in the top right and reports the direction towards North as well as the tilt of the globe.
- Scale Bar - The Scale Bar element is in the lower right corner, and it dynamically reports the general scale of the WorldWind globe.
- View Controls - The View Controls element is in the bottom left corner, and this allows users to use on screen buttons to change the camera incrementally.
- Layer Menu - This menu located on the left allows users to toggle visibility for different layers within the WorldWind Application. This menu is dynamically updated when a new KML or KMZ file is added to the application.



Figure 2 – Configuration of controls on the WorldWindUI Canvas when started up as WxOps WorldWind Earth (WWE).

3.1 WorldWind Canvas Navigation

3.1.1 Mouse Controls

NASA WorldWind natively supports mouse controls to manipulate the globe. The following is a list of function actions for most computer systems. These functions can also be accessed via the onscreen buttons located on the bottom left of the screen (section 3.1.3).

- Left Click on Globe Moves globe and the camera view coordinates so that the clicked location is centered on screen
- Left Click Drag Spins or “pans” the Globe
- Right Click Drag Changes the Roll and Tilt of the camera view
- Mouse Wheel Causes the camera to zoom in or out, changing the altitude of the globe.

3.1.2 Touch Controls

WorldWind also natively supports touch controls for touch-enabled devices.

- Touch on Globe no action (Microsoft Surface)
- Touch and Drag Spins or “pans” the Globe.
- Hold and Drag Changes the Roll (left/right) and Tilt (up/down) of the camera view.

3.1.3 View Controls



Pan left/right and up/down



Zoom in (+) and zoom out (-)



Rotate clockwise and counterclockwise



Tilt up and tilt down



Change “eye” altitude up (+) and down (-)

3.2 Camera Manipulation

The WorldwindUI is provisioned with a JPanel that allows users to input coordinates and camera settings to control the viewing angle and preferences, aka Point Of View (POV). This interface can be accessed from the Menu using "View > Camera Controls" or by pressing [alt+3]. Camera Controls reports the camera settings when users click the "Get Camera" button. To manipulate the camera through this interface, enter the coordinates/settings desired into the text boxes and click the "Set Camera" button. The default camera settings are:

Default Camera Control Input

- Long: -75.00000
- Lat: 38.00000
- Alt: 19070000.0
- Roll: 0.000
- Tilt: 0.000
- Azi: 0.000
- F.O.V.: 45.000

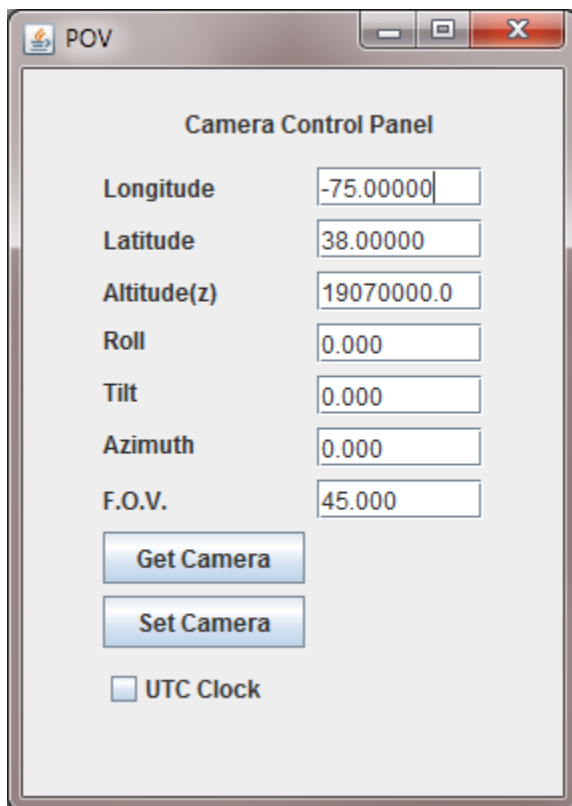


Figure 3 – Camera/POV controls supported by the CameraControlWindow extension.

3.3 KML Support

The WorldWindUI has a menu allowing for importing KML and KMZ files. This can be accessed through Menu item File. The user can choose either “Open KML File” or “Open KML Url.” Alternatively, “Open KML File” can be accessed by pressing [alt+k] and “Open KML Url” can be accessed by pressing [alt+1] or [alt+2]. “Open KML File” will open a file chooser, defaulting to the user’s Documents folder. “Open KML Url” will open a popup allowing the user enter the Url of a KML or KMZ file. Example data are provided for testing at <http://wxops.com/demo/> Defaults are preloaded for convenience:

- [alt+1] <https://wxops.com/demo/Dolly.kmz> = test data with TimeSpans (for Animation)
- [alt+2] <https://wxops.com/demo/WWE-11.kmz> = test data without TimeSpans (Placemark, GroundOverlay, COLLADA Model)

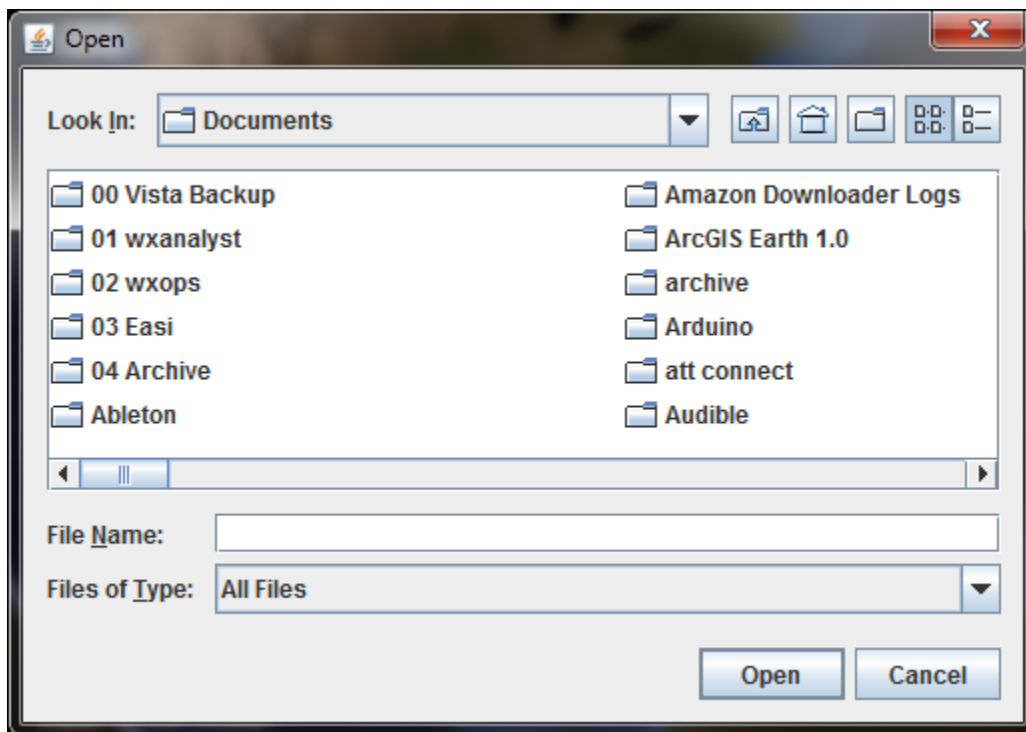


Figure 4 – Open KML File dialog [alt+1].

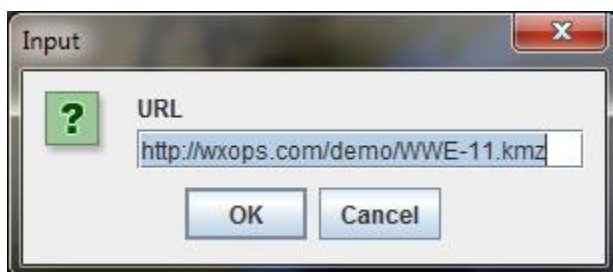


Figure 5 – Open KML Url dialog, opening with a preset default Url [alt+2] for ease of use.

3.4 Animation

The WorldWindUI has elements to start and stop a time animation when any KML or KMZ file contains a TimeSpan element. Unlike other geobrowsers such as Google Earth, the current displayed Valid Time can be set to any value, regardless of the TimeSpans for any layers that are loaded and/or visible. This interface can be accessed from the Menu using "View > Animation Controls" or by pressing [alt+5].

For animation to show data on the Canvas, the KML or KMZ file to be animated must have its TimeSpan "Begin" and "End" values match the corresponding values assigned to the variables "start" and "end". These time variables can be input into the corresponding text fields. Worldwind Earth has its default animation timestamps and delta set to allow for the demo animation of "Hurricane Dolly" [alt+1].

Time information must match the following format:

[year]-[MONTH]-[day]T[hour]:[minute]:[second]Z

Example: 2008-07-23T18:02:00Z

For Hurricane Dolly test KMZ, enter the following:

- Date Begin: 2008-07-23T18:02:00Z
- Date End: 2008-07-23T18:56:00Z
- Time Delta: 6 [minutes]

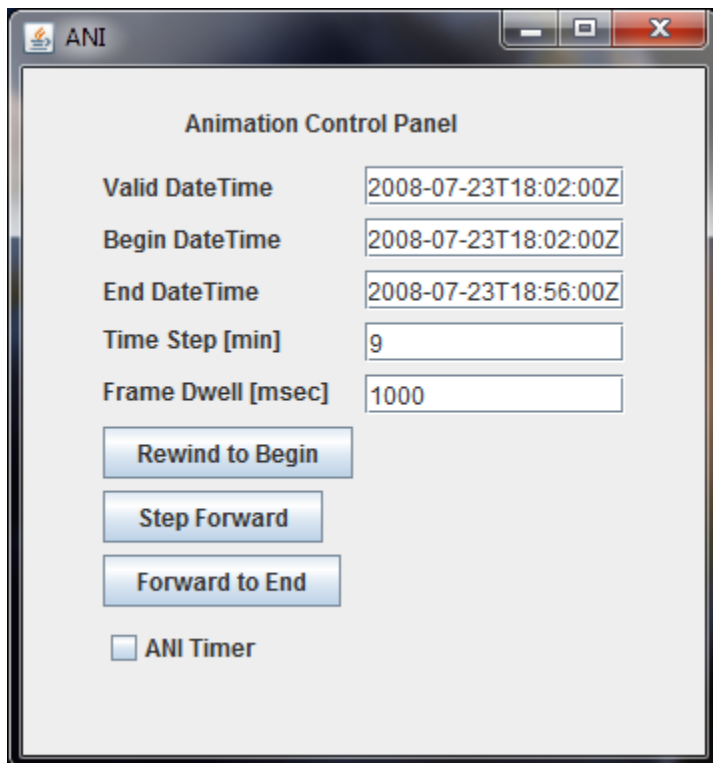


Figure 6 – Animation controls supported by the AnimationControlWindow extension.

4. COM API Emulation

The COM API emulation is currently initiated when the user invokes Camera Controls. This is an advanced topic beyond the scope of this introduction. Please refer to the user manual embedded in the github repository. Current EarthLib IApplicationGE functions emulated are GetCamera() and SetCamera().

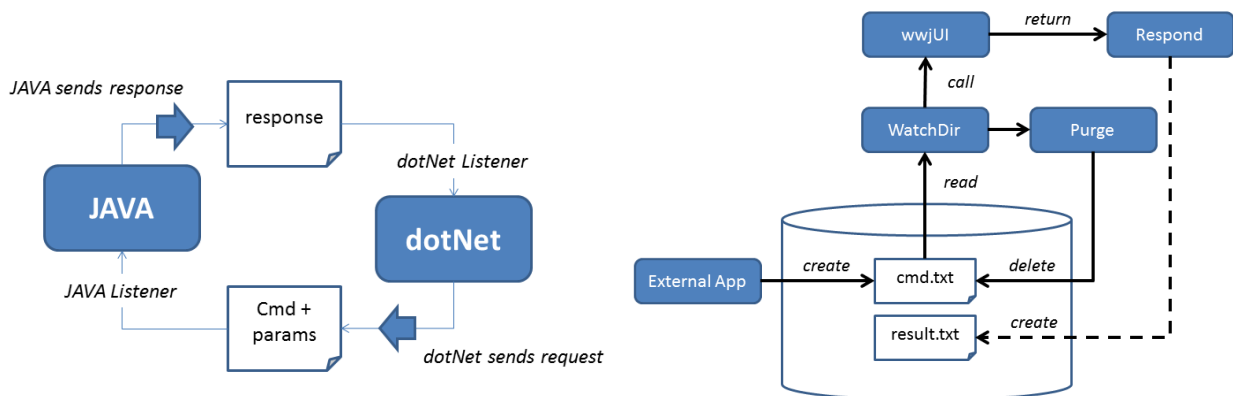


Figure 7 - (left) COM API test with JAVA Listener. (right) Detailed diagram of WatchDir implementation for COM API equivalence in WWJ prototypes.

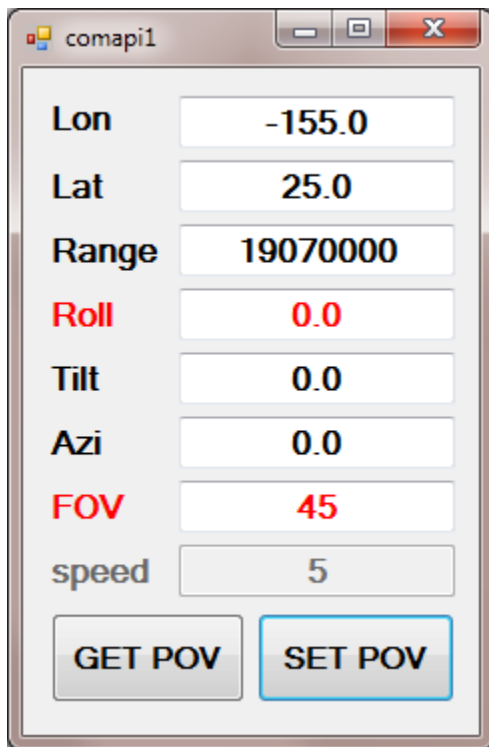


Figure 8 – dotNet sample project (open source) for Visual Studio 2013 Community Edition.

WWE 0.6.2 has implemented the getCamera request as “getpov”, with CameraControlWindow returning the current eye position as POV as a comma delimited string in the format

getpov,lon,lat,range,roll,tilt,azi,fov,speed

Similarly, the setCamera request is implemented as “setpov”, with CameraControlWindow interpreting the current eye position from a POV string provided in the format

setpov,lon,lat,range,roll,tilt,azi,fov,speed

IApplicationGE controls for drawing and KML loading/deletion have been designed. Sidebar database controls are in conceptual design.

EARTHLib.IApplicationGE	F1	ANI	NAV	Panel	Draw	WxAzygy
EARTHLib.IApplicationGE. getCamera			x			x
EARTHLib.IApplicationGE.GetFeatureByHref						
EARTHLib.IApplicationGE.GetFeatureByName						
EARTHLib.IApplicationGE.GetHighlightedFeature						
EARTHLib.IApplicationGE.GetLayersDatabases						
EARTHLib.IApplicationGE. GetMainHwnd	x					
EARTHLib.IApplicationGE. GetMyPlaces	x			x		
EARTHLib.IApplicationGE. GetMyPlaces .GetChildren				x		
EARTHLib.IApplicationGE. GetPointOnTerrainFromScreenCoords					x	x
EARTHLib.IApplicationGE. GetRenderHwnd	x					x
EARTHLib.IApplicationGE. GetTemporaryPlaces				x		
EARTHLib.IApplicationGE. GetTemporaryPlaces .GetChildren				x		
EARTHLib.IApplicationGE.HideDescriptionBalloons						
EARTHLib.IApplicationGE.IsInitialized						
EARTHLib.IApplicationGE.IsOnline						
EARTHLib.IApplicationGE.LoadKmlData						
EARTHLib.IApplicationGE.Login						
EARTHLib.IApplicationGE.Logout						
EARTHLib.IApplicationGE. OpenKmlFile	x	x			x	x
EARTHLib.IApplicationGE.SaveScreenshot						
EARTHLib.IApplicationGE. setCamera			x			
EARTHLib.IApplicationGE.SetCameraParams						
EARTHLib.IApplicationGE.SetFeatureView						
EARTHLib.IApplicationGE.ShowDescriptionBalloon						
EARTHLib.IAnimationControllerGE						
EARTHLib.IAnimationControllerGE.Pause						
EARTHLib.IAnimationControllerGE.Play						
EARTHLib.IAnimationControllerGE. SliderTimeInterval		x				
EARTHLib.IAnimationControllerGE. SliderTimeInterval .BeginTime.Year		x				
EARTHLib.IAnimationControllerGE. CurrentTimeInterval		x				

Figure 9 - Status of COM API Emulation. Navigation controls for getCamera and setCamera are demonstrated (green).

- Green** **Demonstrated**
- Magenta** **In development**
- Yellow** **Designed**
- Cyan** **In design**

5. Contacts



Dr. Scott T. Shipley, WxOps CIO, sshipley@wxops.com
c/o WxOps, Inc. Florida Operations
365 Lanternback Island Drive
Satellite Beach, FL 32940
Office (321) 208 8947
Mobile (571) 309 6024

Peter Subacz, psubacz@wxops.com

6. Acronyms

COE Common Operating Environment
EFB Electronic Flight Bag
KML Keyhole Markup Language
KMZ compressed KML (zip)
POV Point Of View
WWE WorldWind Earth

7. Acknowledgments

NASA SBIR Phase 1 Contract NNX16CA30P
Dr. Richard H. Mogford, NASA COTR, Ames Research Center
Development Team: Paul David Collins, Peter Subacz, Vanessa Haley, Scott Shipley
WxOps Conceptual Guides: Mark Spence, Albert Peterlin
Hawaiian Airlines