NOTICE: This document contains references to Agilent Technologies. Agilent's former Test and Measurement business has become Keysight Technologies. For more information, go to **www.keysight.com.**

KEYSIGHT TECHNOLOGIES

Agilent Technologies

EMPro 2011.01
January 2011
Python Scripting with EMPro

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Java™ is a U.S. trademark of Sun Microsystems, Inc. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", http://www.cs.umn.edu/~metis , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, http://www.intel.com/software/products/mkl

SuperLU_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Python Scripting

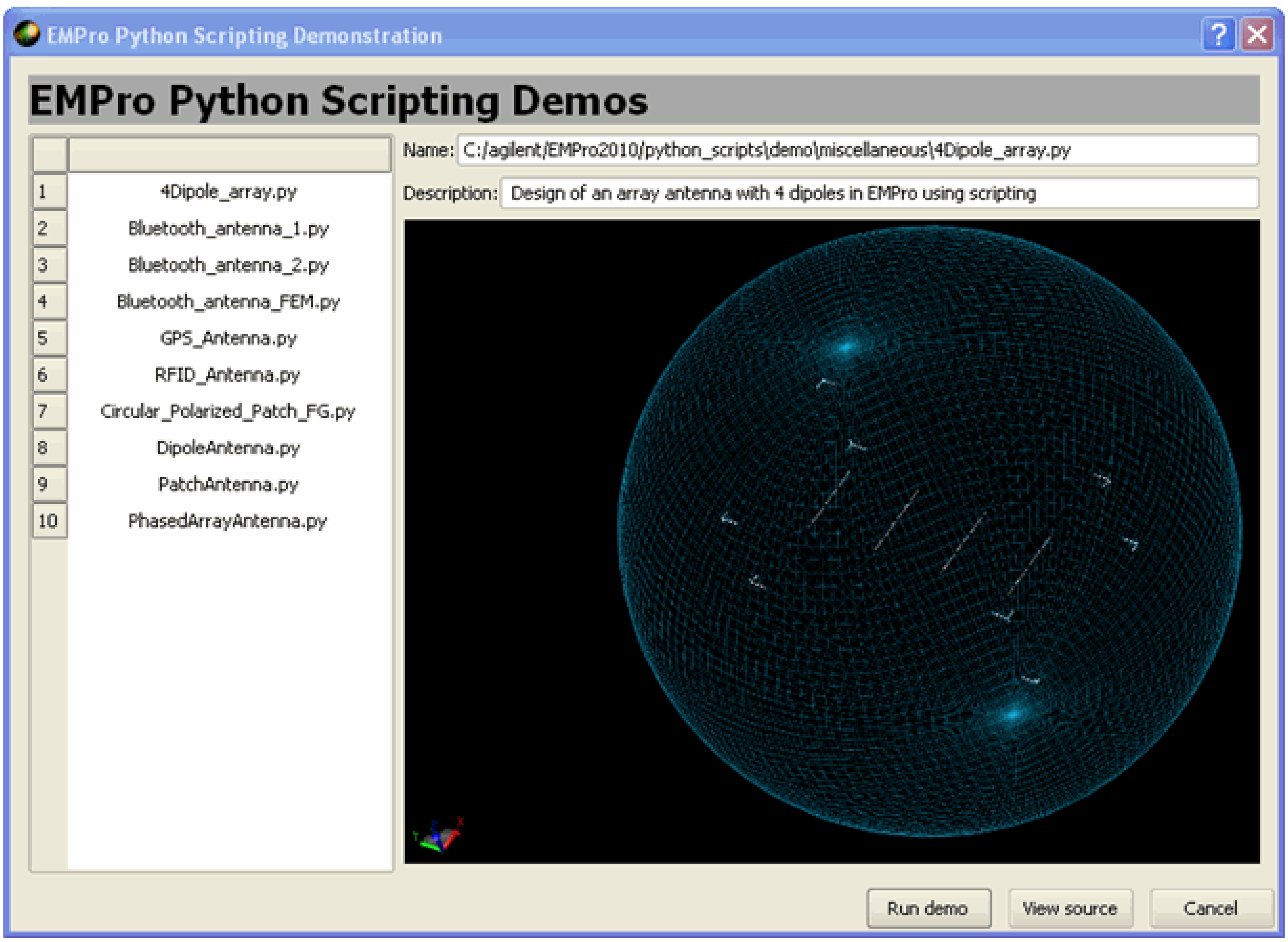
EMPro provides the **Scripting** feature for creating and simulating a project automatically.

Python Demonstration Scripts

In EMPro, Python is used as the language for scripting. To demonstrate this feature, demo scripts are included in EMPro that enables you to set up an EMPro project through script. In addition, a script controls all the important steps of a project creation, such as geometry creation, port definition, mesh and material assignment, boundary definition, near and far field sensor definition, project save, setting up and starting simulation.

Accessing Python Scripts

To access the scripts, choose **EMPro** > **Help** > **Scripting Demo**. This opens the *EMPro Python Scripting Demos* window, as shown in the following figure:



Running Python Scripts

In the EMPro Python Scripting Demos window, select a script on the left side of the *EMPro Python Scripting Demos* window and click **Run Demo** to run a script.

1	4Dipole_array.py
2	Bluetooth_antenna_1.py
3	Bluetooth_antenna_2.py
4	Bluetooth_antenna_FEM.py
5	GPS_Antenna.py
6	RFID_Antenna.py
7	Circular_Polarized_Patch_FG.py
8	DipoleAntenna.py
9	PatchAntenna.py
10	PhasedArrayAntenna.py

Viewing Python Scripts

To view the code of each script, select the required script in the left pane and then click **View Code**. The following scripts are included in EMPro:

- **4Dipole_array.py**: This script creates a project setup to simulate an array of 4 dipole antennas. The individual elements are same that is used in DipoleAntenna.py. The script shows the process of creating repetitive dipoles and ports. Absorbing boundary condition is used in all the directions. The radiation pattern of the array can be seen when this script is run and simulation is over.

- **Bluetooth_antenna_1.py, Bluetooth_antenna_2.py, and Bluetooth_antenna_FEM.py**: These scripts create a project setup to simulate WLAN/Bluetooth and ultra wideband (UWB) applications. Bluetooth antennas designs shown here are small, low-profile, planar microstrip-fed. The designs are taken from IEEE published paper (Source: Bahadir S. Yildirim ,IEEE ANTENNAS AND WIRELESS PROPAGATION LETTERS, VOL. 5, 2006, pp-438)and this demo script shows how to create and simulate bluetooth antenna through script. The antenna exhibits dual-band operation at 2400-2484 MHz (Bluetooth) and 5150-5825 MHz (IEEE 802.11a and HiperLAN/2) bands. The results for the antennas can be seen from result window on the completion of the simulation.

- **GPS_Antenna.py**: This script creates a project setup to simulate a GPS antenna. The script will simulate the radiation pattern at 2000Mhz according to the IEEE 802.11b standard. Antenna consists of an S-strip and a T-strip which are separately printed on the two sides of a thin substrate.In the structure bottom which is S strip acts as a ground for the pcb board on which antenna geometry is created. The other side which is T strip is radiating element.The complex geometry is completly generated through script automatically along with assigning port,material and grids.

The S parameter and far field radiation pattern for the GPS antenna can be seen when this script is run and simulation is complete.

- **RFID_Antenna.py**: This script creates a project setup to simulate RFID handheld readers. The antenna consists of a microstrip-to-coplanar stripline transition, a meandered driven dipole, a closely-coupled parasitic element, and a folded finite-size ground plane. The complex geometry is created through script along with port and boundary assignment. The resonating frequency of antenna is 945 MHz and results can be seen when this script is run and simulation is over.

- **Circular_Polarized_Patch_FG.py**: This script creates a project setup to simulate a circular polarized patch antenna where geometry is perturbed to achieve circular polarization. The complex perturbed geometry of the antenna is automatically created though script. The finite ground plane for the patch geometry is created and absorbing boundary condition is assigned in all the directions automatically on the run of the script. The S parameter, far field radiation pattern and axial ratio plot for this antenna can be seen in results after the completion of simulation.

- **DipoleAntenna.py**: This script creates a project setup to simulate a dipole antenna, which is operating at 0.47GHz. In this script, the wire element is used to draw the dipole and port is placed on wire body to excite it. Absorbing boundary condition is used in all the directions.On the run of the script, simulation starts. In simulation, through script steady state data is collected at 0.47 GHz. The S parameter for wide band and 0.47 GHz frequency as well as far field radiation pattern for the dipole antenna can be seen in results when this script is run and simulation is over.

- **PatchAntenna.py**: This script creates a project setup to simulate a patch antenna which is operating in C band. The patch is a planar geometry and scripts shows how planar objects like quadplate for patch along with 3D objects like box for substrate is created. Patch antenna needs ground plane implemented in project through infinite PEC boundary condition along with zero padding in grid at lower Z set through script. The S parameter and far field radiation pattern for the Patch antenna can be seen when this script is run and simulation is over.

- **PhasedArrayAntenna.py**: This script creates a project setup to simulate a phased array antenna operating at 1.329 GHz. When this script is run, you need to specify the number of elements and phase difference. When this input is provided to the script the geometry for the antenna for the specified number of elements will be created automatically. The Phase difference will be assigned to respective ports and simulation will start automatically. Once the simulation is completed the the angular shift in main beam for the provided phase difference can be seen from 2D and 3D radiation pattern from result window.