**INTRODUCTION:**

Signal integrity analysis using ANSYS SIWave often involves a significant amount of manual work, which can be time-consuming and prone to errors. This manual provides a comprehensive guide to automation scripts designed to alleviate these challenges and enhance productivity.

By leveraging automation, users can drastically reduce manual labor and streamline their workflow, resulting in more efficient signal integrity analysis. Whether you're a seasoned engineer or a PCB designer, these scripts offer a solution to the repetitive tasks involved in SIWave analysis.

This guide covers everything you need to know to get started with the automation scripts. From installation and basic concepts to advanced customization options and troubleshooting tips, this manual should set a solid foundation. By the end of this manual, you'll have a solid understanding of how to harness the automation to optimize the signal integrity analysis process.

**GETTING STARTED:**

There are 4 main scripts regarding SIWave simulation, extraction, and reporting which are listed down below:

* Automation for 1-DPC and 2-DPC (1-DPC\_Automation.py & 2-DPC\_Automation.py)
* Automation for High-Low-Typical Testing (CornerModeling.py)
* Automation for Editing Port Names (TouchstoneHeader.py)
* Automation for Reporting (SIWaveReporting.py)

However, there a multitude of dependencies and libraries that are necessary for the Python scripts to work as intended. First, Python must be installed on your computer. Check out the Python installation guide [here](https://www.howtogeek.com/197947/how-to-install-python-on-windows/). Afterwards, ensure that pip is installed to download other dependencies. With the Windows PowerShell terminal, execute the command “py -m ensurepip --upgrade”, which will download the pip tool for library installation. Afterwards, run the command “pip install -r requirements.txt” and make sure that requirements.txt is in the same directory. This will download all the packages necessary for the Python scripts.

Afterwards, download and install Visual Studio Code or an IDE for easy editing and viewing of the Python scripts. The link for Visual Studio Code is [here](https://code.visualstudio.com/). Furthermore, ensure that ANSYS AEDT and SIWave are properly installed.

**BASIC CONCEPTS:**

There are four primary scripts central to SIWave simulation, extraction, and reporting:

* Automation for 1-DPC and 2-DPC: (1-DPC\_Automation.py & 2-DPC\_Automation.py)
* Automation for High-Low-Typical Testing: (CornerModeling.py)
* Automation for Editing Port Names: (TouchstoneHeader.py)
* Automation for Reporting: (SIWaveReporting.py)

However, to ensure the seamless functionality of these Python scripts, it's imperative to address their dependencies and requisite libraries.

Firstly, Python must be installed on your system. Refer to the Python installation guide under **References & Resources** section for detailed instructions. Subsequently, verify the presence of pip, the Python package installer, which facilitates the acquisition of essential dependencies.

Utilizing the Windows PowerShell terminal, execute the command **py -m ensurepip --upgrade** to ensure the latest version of pip is installed. Following this, execute **pip install -r requirements.txt**, ensuring that the **requirements.txt** file resides in the same directory. This action will prompt the download and installation of all requisite Python packages.

Furthermore, for efficient editing and viewing of the Python scripts, it's recommended to utilize an Integrated Development Environment (IDE) such as Visual Studio Code. The download link for Visual Studio Code and other essential resources and references can be found in the under **References & Resources** section.

Additionally, it's crucial to ensure that ANSYS AEDT and SIWave are correctly installed to facilitate seamless integration with the Python scripts.

**USER INTERFACE:**

Each script features an intuitive user interface designed to streamline workflow effortlessly. Whether facilitating manipulation of ANSYS SIWave or providing straightforward GUIs for reporting files or renaming components, these interfaces are tailored to enhance usability.

The usage instructions outlined in the following section are crafted to ensure that the GUIs are readily understandable, and the step-by-step guidance is straightforward to follow. By adhering to these instructions, users can navigate the interfaces seamlessly, maximizing the efficiency and effectiveness of their tasks.

**USAGE INSTRUCTIONS:**

For the **1-DPC and 2-DPC Automation scripts**, the steps are the following:

1. **Accessing SIWave Tools:** Launch the latest version of SIWave and navigate to the Tools menu. Select "Run Script" located on the far right.
2. **Selecting Automation Script:** Choose the appropriate Python script corresponding to the desired DPC analysis, whether it be for 1-DPC or 2-DPC.
3. **Configuring Channel Extraction:** Identify and select the target .siw file from which channel extraction is required. Specify the destination folder for storing the resulting files and proceed to the next step.

**A computer screen shot of a computer screen

Description automatically generated**

1. A screenshot of a computer

   Description automatically generated**Selecting Channels:** Identify the specific nets requiring extraction and input their names into the designated field. Append as many net sets as necessary to encompass all desired channels for analysis.
2. A screenshot of a computer

   Description automatically generated**Layer Stackup Configuration:** Define the layer stackup parameters by assigning specific Huray models (High, Medium, or Low Loss) to relevant layers. Additionally, create a new material with its properties by specifying frequency, permittivity, tangent loss, and select the desired conductor material.
3. **Die Stack Customization:** Tailor the die stack configuration by indicating the discrete devices to be modified. Ensure adherence to the prescribed naming convention, prioritizing the APU designation before specifying the DIMM.

A screenshot of a computer

Description automatically generated

1. A screenshot of a computer

   Description automatically generated**SYZ Parameters Setup:** Configure the SYZ (S-parameter) analysis parameters by defining the start and stop frequencies, along with the desired step size. This step ensures comprehensive frequency coverage for accurate simulation results.

For the **Corner Modeling script**, the steps are the following:

1. **Accessing SIWave Tool**s: Launch the latest version of SIWave and navigate to the Tools menu. From there, select "Run Script" located on the far right.
2. **Selecting Corner Modeling Script:** Choose the Python script titled CornerModeling.py, tailored for corner modeling operations.
3. **Choosing Folder:** Specify the directory containing all typical files pertinent to the analysis.
4. **Selecting Stackup File:** Choose the stackup file (.stk) utilized during the generation of typical files to ensure consistency in simulation parameters.
5. **Material Selection:** From the designated box, choose the material requiring modification.
6. A screenshot of a computer

   Description automatically generated**Inputting Material Parameters:** Input the desired high permittivity and high loss tangent values, as well as the low permittivity and low loss tangent values, into the respective boxes. Ensure accuracy to maintain simulation integrity.

For the **Touchstone Header script,** the steps are the following:

1. **Opening Touchstone Header Python File:** Navigate to the Touchstone Header executable Python file and click to open it.
2. **Specifying Input Directory:** Choose the directory containing the Touchstone files that require editing. This directory serves as the input location for the script.
3. A screenshot of a computer

   Description automatically generated**Setting Output Directory:** Specify the directory where you want the edited Touchstone files to be generated. This directory will store the modified files produced by the script.

For the **SIWave Reporting script,** the steps are the following:

1. **Launching SIWave Reporting Executable Python File:** Locate and click on the SIWave Reporting executable Python file to open it.
2. **Selecting Folder Containing Touchstone Files:** Specify the directory containing the Touchstone files that require reporting. This folder serves as the input location for the reporting script.
3. A screenshot of a computer

   Description automatically generated**Automatic PowerPoint Generation:** Upon execution, the script will automatically generate a PowerPoint presentation named "test.pptx". This presentation will showcase insertion loss and return loss data extracted from the specified Touchstone files.

**NOTE:** For the **Touchstone Header & SIWave Reporting scripts**, the user has two options in running the script. One being through the terminal or through making an executable python file. The tutorial for converting the script into an executable file is the following:

1. **Creating Shortcut:** Right-click on the Python script file and select "Create Shortcut."
2. **Accessing Properties:** Right-click on the newly created Shortcut file and select "Properties."
3. **Modifying Target Location:** In the "Target" field of the Properties window, prepend "pythonw" to the existing text. This modification allows the script to be executed as an executable file.
4. **Applying Changes:** Click "Apply" to save the changes. The Shortcut file is now converted into an executable file, allowing users to simply click on it to execute the Python script.

**CUSTOMIZATION & CONFIGURATION:**

Should the scripts fail to meet the user's specifications, it's advisable to duplicate the file and integrate the requisite functions by adhering to the SIWave Scripting Guide or manually inspecting the .siw file to discern Python-editable components. Moreover, for users seeking comprehension of the script, meticulous scrutiny of comments and code stack is recommended to grasp the scripts' objectives thoroughly. For more information on resources, look at the References & Resources section of the user manual.

**MAINTENANCE & UPDATES:**

Should the user seek modifications to the original file, it is imperative to maintain the integrity of the original. Instead, create a duplicate of the file, thereby preserving version control. Ensure that this duplicate is meticulously annotated throughout and titled distinctly to obviate confusion. This guarantees that the original remains untouched, safeguarding against inadvertent alterations. Additionally, uphold the practice of creating regular backups of both the original and modified scripts to prevent data loss in case of accidental modifications or system failures.

After implementing changes or updates, it’s crucial to conduct thorough testing to ensure that the modified script behaves as expected and fulfills the user’s requirements. Furthermore, users should maintain comprehensive documentation for any modifications added to the scripts. This documentation should include details such as the purpose of the changes, specific alterations made, and any dependencies introduced. Additionally, update the user manual as needed to reflect any changes in functionality or configuration.

**TROUBLESHOOTING:**

In the event of sudden crashes with the scripts or ANSYS SIWave, it's essential to investigate the log files that develop in the same directory as the .siw file. These log files serve as vital diagnostic tools, highlighting where the script concluded and providing invaluable insights into potential issues.

Common issues may arise from inadvertently clicking on ANSYS SIWave while it executes the script. ANSYS SIWave requires time to generate these channel files, and clicking on it during operation may result in an error message indicating that the application is not running, despite it being active. Therefore, users are strongly advised against interacting with ANSYS SIWave unless explicitly prompted to do so by the script and GUI.

By adhering to this guidance and diligently reviewing the log files, users can effectively identify and address any anomalies or errors encountered during script execution, ensuring a smoother workflow and minimizing disruptions.

**POTENTIAL FUTURE IMPROVEMENTS:**

Several features and enhancements are recommended for further implementation:

1. **SIWave Reporting:** Since this script was created most recently, it may not be as robust as desired. To improve its functionality, rigorous testing of return loss and insertion loss is warranted to ensure accurate results. Additionally, considering that touchstone files and port names may not always be in a consistent order, developing a function to automatically identify and pair them based on their names could streamline the process. Furthermore, exploring the SignalNetAnalyzer.py file for insights on implementing and reporting impedance data could enhance the script's capabilities.
2. **Exploration of Other Scripting Technologies:** It's worth delving into other libraries and automation technologies, such as PyAEDT, to broaden the toolkit and potentially revolutionize existing scripts or develop new ones. Diversifying resources in this manner can lead to innovative solutions and improved efficiency.

**REFERENCES & RESOURCES:**

For further assistance and resources, please refer to:

* Visual Studio Code: A powerful integrated development environment (IDE) for editing and managing Python scripts. ([Visual Studio Code](https://code.visualstudio.com/))
* Install Python on Windows: Step-by-step instructions for installing Python on a Windows operating system. ([Install Python on Windows](https://www.howtogeek.com/197947/how-to-install-python-on-windows/))
* SIWave Help Documentation: Comprehensive resources including SIWave.pdf and SIwaveScriptingGuide.pdf, located in the codebase folder. These documents provide detailed information and guidance on utilizing SIWave and scripting functionalities.
* GitHub Repository: A centralized source containing all scripts and resources for automation tasks. ([AutomationSIPI-GitHub](https://github.com/loganchoi/AutomationSIPI-AMD))

We strongly recommend consulting these resources for additional support and guidance throughout the scripting process.

This addition encourages users to leverage these resources effectively, enhancing their understanding and proficiency in utilizing the scripts and associated tools.