NI Measurement Plug-In Converter

Contents

[Overview 2](#_Toc176361667)

[Related Links 2](#_Toc176361668)

[Software Requirements 2](#_Toc176361669)

[Installation 2](#_Toc176361670)

[Note: 2](#_Toc176361671)

[Supported Data types & Instrument Drivers 3](#_Toc176361672)

[Prerequisites 3](#_Toc176361673)

[Limitations 5](#_Toc176361674)

[Known Issues 5](#_Toc176361675)

[How to convert measurements using NI Measurement Plug-In Converter? 6](#_Toc176361676)

[Additional Steps for VISA instruments 6](#_Toc176361677)

[Event Logger 8](#_Toc176361678)

# Overview

NI Measurement Plug-In Converter is a CLI tool to convert Python measurements to measurement plug-ins.

# Related Links

* [Measurement Plug-In Overview - NI](https://www.ni.com/docs/en-US/bundle/measurementplugins/page/measurement-plugins.html)
* [TestStand Manual](https://www.ni.com/docs/en-US/bundle/teststand/page/user-manual-welcome.html)

# Software Requirements

[Python 3.8.5](https://www.python.org/downloads/release/python-385/)

|  |  |
| --- | --- |
| **Python Packages** | **Version Required** |
| Mako | 1.2.1 or above |
| Click | 8.1.3 or above |
| Astor | 0.8.1 or above |
| Pydantic | 2.8.2 or above |
| ni-measurement-ui-creator | 1.0.0.dev8 or above |
| Black | 24.8.0 or above |

|  |  |
| --- | --- |
| NI Packages | Version Required |
| NI InstrumentStudio Pro | 2024 Q3 or above (Optional) |
| NI TestStand | 2023 Q4 or above (Optional) |

# Installation

* Ensure all the dependency files are placed parallel to `install.bat`
* Run the `install.bat` file by double clicking it.

## Note:

* Require internet access to install the external dependencies of NI Measurement Plug-In Converter.

# Supported Data types & Instrument Drivers

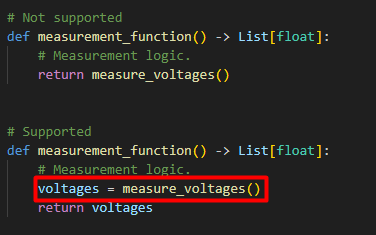
|  |  |
| --- | --- |
| **Data types** | **Instrument Drivers** |
| Int | NI-DCPower |
| Float | NI-DMM |
| Str | NI-Digital |
| Bool | NI-FGEN |
| List[int] | NI-Scope |
| List[float] | NI-Switch |
| List[str]] | NI-DAQmx |
| List[bool] | PyVISA for NI VISA |

* Inputs and outputs of unsupported data types will be skipped.
* Python measurement must use one of the supported instrument drivers.

# Prerequisites

The Python measurement should contain a measurement function which should

* Use one of the [supported drivers and data types](#_Supported_Data_types_1). Inputs and outputs of unsupported data types will be skipped.
* Contain a return value. The return value should be a variable and not a direct function call or constant value.

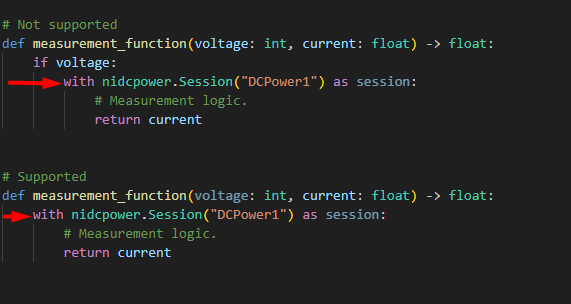


* Have properly type hinted inputs and outputs

A computer screen shot of text

Description automatically generated

* Have the instrument driver's session initialization inside the measurement function and within the next level of indentation.



* Have all the instrument driver's session initialization at a single point using the context manager `with`.

A screen shot of a computer program

Description automatically generated

# Limitations

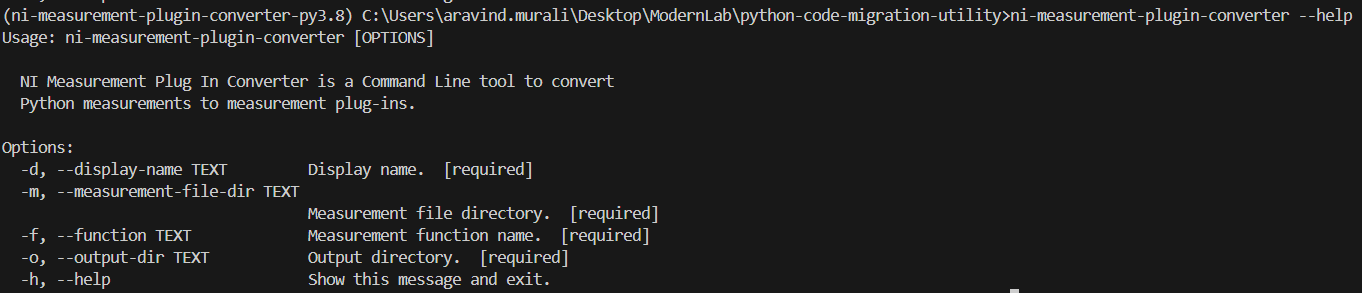
* Conversion of class based measurements are not supported.
* Measurement UI generated by the tool will not include controls and indicators for lists of strings and lists of booleans for their respective inputs and outputs.

# Known Issues

* Measurements which don’t follow the prerequisites will not be converted correctly.
* For measurements that use VISA instruments, a few more additional steps must be followed. Refer [Additional Steps for VISA instruments](#_Additional_Steps_for_2)

# How to convert measurements using NI Measurement Plug-In Converter?

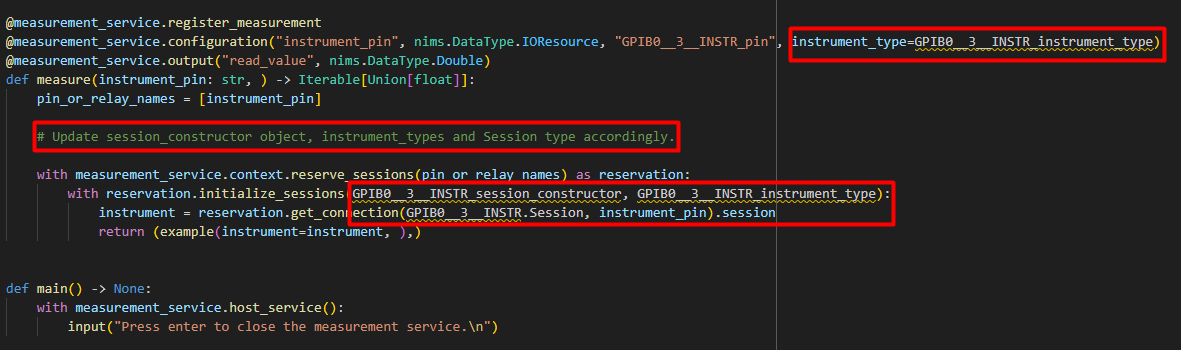
* Install the NI Measurement Plug-In Converter. Refer [Installation](#_Installation).
* Open Command Prompt.
* Run the below command,
  + ni-measurement-plugin-converter --display-name <Display Name> --measurement-file-dir <Measurement file directory> --function <Measurement function name> --output-dir <Output directory>



* Modify the CLI arguments’ values with appropriate values.
* Measurement plug-in will be created at the output directory.

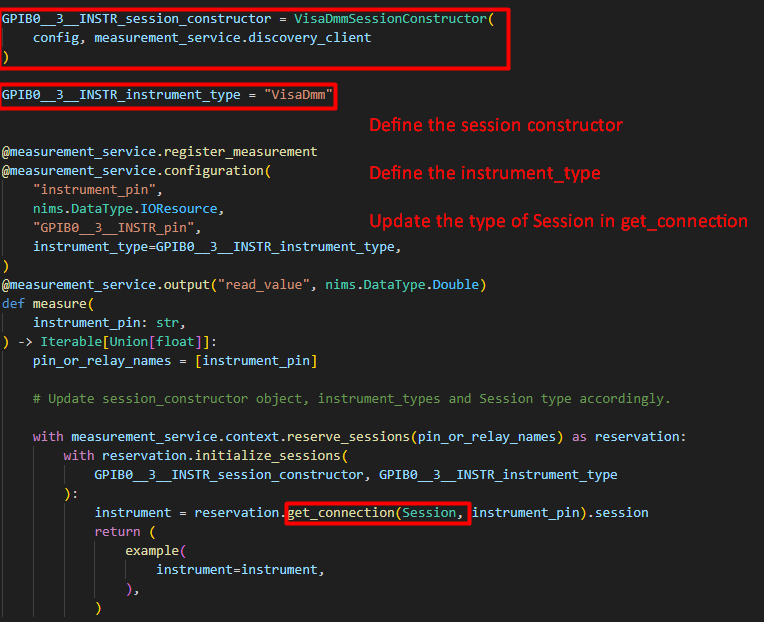
# Additional Steps for VISA instruments

For measurement that use VISA instruments, the session\_constructor, session type and instrument\_types must be updated with appropriate values.



Steps to be followed

* Define the grpc support.
* Define the Session class for the instrument type.
* Define the session constructor for the instrument type.



* For session\_constructor, create SessionConstructor object of the instrument driver used.
* For instrument\_type, use the pin map instrument type id.
* For session type, the type of session should be passed.

For details, refer [Example VISA](https://github.com/ni/measurement-plugin-python/tree/releases/2.0/examples/nivisa_dmm_measurement)

A screenshot of a computer program

Description automatically generated

# Event Logger

* The tool will generate a log once the conversion process is started, documenting all the actions performed by the tool throughout the conversion process.
* Log file can be found at the output directory.
* The log includes the details about any errors encountered during the conversion process.