

Python Integration in JMP

Changes in JMP 18

Martin Demel



Extend your JMP Experience



Python Library embedded

- Simplified Access to Python
- Better Maintenance
- Python Script Editor
 - `Dialog and Regression.py`
- Extended Capabilities



Data Connectors

- Connect to special data sources
- e.g. Apache Parquet Files

Python Connection

Before and after JMP 18

JMP 17 and older (only through JSL)

JMP 18 (through JSL OR direct in JMP's new  Editor)

Python Connect

Syntax: PythonConnect(<Echo(boolean)>, <Path(Path to Python DLL or shared library)>, <Use Python Version(version string)>, <Python sys path(JSL list of paths defining a Python sys path set)>)

Version Added: 14

Description: Returns a Python connection scriptable object.

Python Init

Syntax: Python Init(<Echo(boolean)>, <Path(Path to Python DLL or shared library)>, <Use Python Version(version string)>, <Python sys path(JSL list of paths defining a Python sys path set)>)

Version Added: 14

Description: Initializes the Python Interfaces.

Python Submit

Syntax: Python Submit(statements)

Version Added: 14

Description: Submits statements to Python. Statements can be in the form of a string value or list of string values.

Python Get

Python Execute

Python Send

...

Example

```
Names Default To Here( 1 );
Python Init();
Python Submit(
    "\[
str = 'The quick brown fox jumps over the lazy dog';
]\\"
);
getStr = Python Get( str );
Show( getStr );
Python Term();
```

Example 1

```
import jmp

import jmputils
# update to latest version of pip and setuptools then install numpy & pandas
jmputils.jpip('install --upgrade', 'pip setuptools')
jmputils.jpip('install', 'numpy pandas')
```

Why Changed?

Customer challenges JMP 14-17

- **#1 Issue reported – Just getting Python support to work.**
 - If everything went fine, it just worked... all too often it didn't.
 - Numerous Python distributions and environments, made sufficient testing impossible.
 - Anaconda could be made to work on Windows, *sometimes*.
 - Python virtual environments could be made to work *sometimes*.
- **Inconsistent environment from machine to machine.**
 - Script invocation could not depend even on a common Python version in the user's environment.
- **JSL editor is unfriendly to Python.**
 - Coding Python as a JSL string was difficult, and unpleasant.
- **Insufficient error reporting from Python.** Difficult for users to find location Python of errors.
- **Data Table / Matrix transfer through file system is *slow* for big data.**

Vision

- ✓ Python Integration just works, no configuration.
 - Isolated Python 3.11.x is embedded and installed as part of JMP.
- ✓ Environment common for all users of the same JMP version.
- ✓ In-memory data transfer and zero-copy memory buffer access.
- ✓ Inviting and productive environment for Python programmers.
 - Code Coloring / Syntax Highlighting of Python scripts.
 - Run pure Python scripts from the script editor.
 - Access JMP data, log, JSL and classes from Python. `import jmp`
 - Greatly enhanced error reporting to the JMP log and embedded log.
- ✓ Tested and verified.
 - Our environment is the same as user environment.
- ✓ Scripting Index support for Python category
- ✓ User ability to extend JMP's capabilities with Python's vast library of packages.
- ✓ This is not an attempt at creating a 1:1 alternative to JSL. Instead, a familiar and productive environment for Python programming within JMP.

Python related Presentations

at DSE 2024, Manchester

Paul Nelson, JMP:

- [JMP-and-Python-The-Next-Step \(Discovery Summit Europe 2024, Manchester\)](#)
- [Developer Tutorial: New Python Integration and Scripting Capabilities in JMP 18](#)

User Presentations:

- Bundesdruckerei: [A-Case-Study-of-JMP-Live-Integration-with-Python-for-Production](#)
- Rentschler: [Integrating-JMP-Data-Exploration-and-Python-Machine-Learning](#)

Learn more

The screenshot shows the JMP Online Help & Scripting Index page for Python. The page title is "Python" and the subtitle is "Extend JMP Capabilities with the Integrated Python Environment". The main content area includes a brief introduction: "JMP includes a robust Python integration feature set, combining Python's versatility with JMP's data visualization and analysis capabilities." Below this, a list of Python integration features is provided:

- A Python script editor available within JMP
- Easy extension of Python with packages such as Pandas and PyArrow
- PY file support
- Enhanced Error Log reporting
- Direct memory access from JMP data tables and JSL to Python

Below the list, there is a section titled "Figure 15.1 Python Script Editor with Embedded Log" which shows a screenshot of the Python script editor interface. The code in the editor is as follows:

```
1 import jmp
2
3 # Create a data table
4 # dt = jmp.DataTable(name='table_name', rows=n)
5 pbp = jmp.DataTable(rows=5)
6 pbp.name = 'Powered by Python'
7 pbp.new_column('Name', jmp.DataType.Character)
8 pbp.new_column('Hourly Rate')
9 #
10 pbp['Name'] = ['Janet', 'James', 'Jerry', 'Jenny', 'Jill']
11 pbp['Hourly Rate'] = [14.25, 9.75, 15.0, 12.25, 17.25] # last value had no comma, missing
```

<https://community.jmp.com/>

The screenshot shows the JMP User Community search page for Python. The page title is "Search - JMP User Community" and the URL is "community.jmp.com/t5/forums/searchpage/?q=python&noSynonym=false&collapse_discussion=true". The page features a search bar with the text "python" and a "Turn on suggestions" button. Below the search bar, there is a large magnifying glass graphic with the text "Search" and "Search the Community". At the bottom of the page, there are navigation links for "POSTS", "USERS", "PRIVATE MESSAGES", "PLACES", and "PRODUCTS".

[Online Help & Scripting Index](https://www.jmp.com/support/help/en/18.0/#page/jmp/python.shtml#)

<https://www.jmp.com/support/help/en/18.0/#page/jmp/python.shtml#>