# PyMechanical cheat sheet

Ansys

**Version: 0.11.1  (stable)**

## A. Connect to a remote session of Mechanical from Python

### Launch and connect to a session

Launch and connect to Mechanical locally:

```
import ansys.mechanical.core as pymechanical

mechanical = pymechanical.launch_mechanical()
```

Launch Mechanical from a local or remote terminal:

```
# Standalone Mechanical from a local or remote
    terminal
ansys-mechanical -r 242 --port 10000 -g
```

Manually connect to this Mechanical session from a local client:

```
import ansys.mechanical.core as pymechanical
# #Note: The following code uses port 10000, but you
    can specify an alternative port if required.

# Either connect locally
mechanical = pymechanical.Mechanical(port=10000)

# Or connect remotely, specifying the IP address or
    hostname and port.
mechanical = pymechanical.Mechanical("192.168.0.1",
    port=10000)
```

### Launch by version

Verify the license and version of Mechanical that is used:

```
print(mechanical)
```

Launch a specific version of Mechanical:

```
from ansys.mechanical.core import find_mechanical

wb_exe = find_mechanical(242)[0]
# 'Ansys Inc\\v242\\aisol\\bin\\winx64\\AnsysWBU.exe'
mechanical = launch_mechanical(
    exec_file=wb_exe, verbose_mechanical=True, batch=
        True)
print(mechanical)
```

### Launch the Mechanical UI

Launch the Mechanical UI:

```
mechanical = pymechanical.launch_mechanical(batch=
    False)
```

## Send commands to Mechanical

Run a single command:

```
result1 = mechanical.run_python_script("2+3")
result2 = mechanical.run_python_script(
    "ExtAPI.DataModel.Project.ProjectDirectory"
)
mechanical.run_python_script(
    "Model.AddStaticStructuralAnalysis()"
)
```

Execute a block of commands:

```
# Import a  material
commands = """
cu_mat__file_path = r'D:\Workdir\copper.xml'.replace
    ("\\", "\\\\")
materials = ExtAPI.DataModel.Project.Model.Materials
materials.Import(cu_mat__file_path)"""
mechanical.run_python_script(commands)
```

Execute a Python script file:

```
mechanical.run_python_script_from_file(file_path)
```

Import a Mechancial file and print the count of bodies:

```
file = r"D:\\Workdir\\bracket.mechdb"
command = f'ExtAPI.DataModel.Project.Open("{file}")'
mechanical.run_python_script(command)
mechanical.run_python_script(
    "allbodies=ExtAPI.DataModel.Project.Model.
        GetChildren( DataModelObjectCategory.Body,
        True)")
mechanical.run_python_script("allbodies.Count")
```

Perform project-specific operations:

```
# Get the project directory
mechanical.project_directory

# List the files in the working directory.
mechanical.list_files()
# Save
mechanical.run_python_script(
    "ExtAPI.DataModel.Project.Save(r'D:\\Workdir')")
# Log in two ways:
mechanical._log.info("This is a useful message.")
mechanical.log_message("INFO", "info message")
# Exit
mechanical.exit(force=True)
```

## B. Load an embedded instance of Mechanical in Python

Embed a Mechanical instance:

```
from ansys.mechanical.core import App

app = App(version=242)
print(app)
```

Extract and merge global API entry points:

```
# Extract the global API entry points (available from
    built-in Mechanical scripting)
# Merge them into your Python global variables
app.update_globals(globals())
```

Access entry points from Python:

```
ExtAPI   # Application.ExtAPI
DataModel  # Application.DataModel
Model  # Application.DataModel.Project.Model
Tree  # Application.DataModel.Tree
Graphics  # Application.ExtAPI.Graphics
```

Import a file and print the count of bodies:

```
file = r"D:\\Workdir\\bracket.mechdb"
app.open(file)
app.update_globals(globals())
allbodies = Model.GetChildren(DataModelObjectCategory.
    Body, True)
print(allbodies.Count)
```

Turn on warning logging:

```
import logging
from ansys.mechanical.core import App
from ansys.mechanical.core.embedding.logger import (
    Configuration,
    Logger)

Configuration.configure(level=logging.WARNING,
    to_stdout=True)
app = App(version=242)
Logger.error("message")
```

## References from PyMechanical and Mechanical documentation

- Getting started
- Examples
- API reference
- Scripting in Mechanical Guide