



© 2025 ANSYS, Inc. or affiliated companies
Unauthorized use, distribution, or duplication prohibited.

ansys-magnet-segmentation-toolkit



ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
<http://www.ansys.com>
(T) 724-746-3304
(F) 724-514-9494

Mar 31, 2025

ANSYS, Inc. and
ANSYS Europe,
Ltd. are UL
registered ISO
9001:2015
companies.

CONTENTS

The Magnet Segmentation Toolkit is a Python wrapper for automating the segmentation and skew of inner rotor interior permanent magnet (IPM) and surface permanent magnet (SPM) motors using Ansys Electronics Desktop (AEDT). You can launch this toolkit from the AEDT UI or launch it directly from a Python console.

Getting started Learn more about the Magnet Segmentation Toolkit and how to install it.

Getting started **UI reference** Understand how to use the Magnet Segmentation Toolkit Wizard.

UI reference **API reference** Understand the APIs available for the Magnet Segmentation Toolkit.

API reference **Examples** Explore example that shows how to use the Magnet Segmentation Toolkit to perform many different types of operations.

Examples **Contribute** Learn how to contribute to the Magnet Segmentation Toolkit codebase or documentation.

Contribute

GETTING STARTED

This section explains how to install the AEDT Magnet Segmentation Toolkit.

Installation Learn how to install the AEDT Magnet Segmentation Toolkit.

Installation **User guide** Learn more about the Magnet Segmentation wizard and how to use it.

User guide

1.1 Installation

The Magnet Segmentation Toolkit can be installed like any other open source package.

You can either install both the backend and user interface (UI) methods or install only the backend methods.

To install both the backend and UI methods, run this command:

```
pip install ansys-magnet-segmentation-toolkit[all]
```

If you only need the common API, install only the backend methods with this command:

```
pip install ansys-magnet-segmentation-toolkit
```

To install the toolkit offline, you can use a wheelhouse. On the [Releases](#) page, you can find the wheelhouses for specific release in its asserts and download the wheelhouse.

You can then install the toolkit with this command:

```
pip install --no-cache-dir --no-index --find-links=<path_to_wheelhouse>/ansys-magnet-  
↪segmentation-toolkit-v0.3.3-wheelhouse-windows-latest-3.10 ansys-magnet-segmentation-  
↪toolkit
```

You can also install the toolkit using the toolkit manager. For more information, see the toolkit manager (TBD).

1.2 User guide

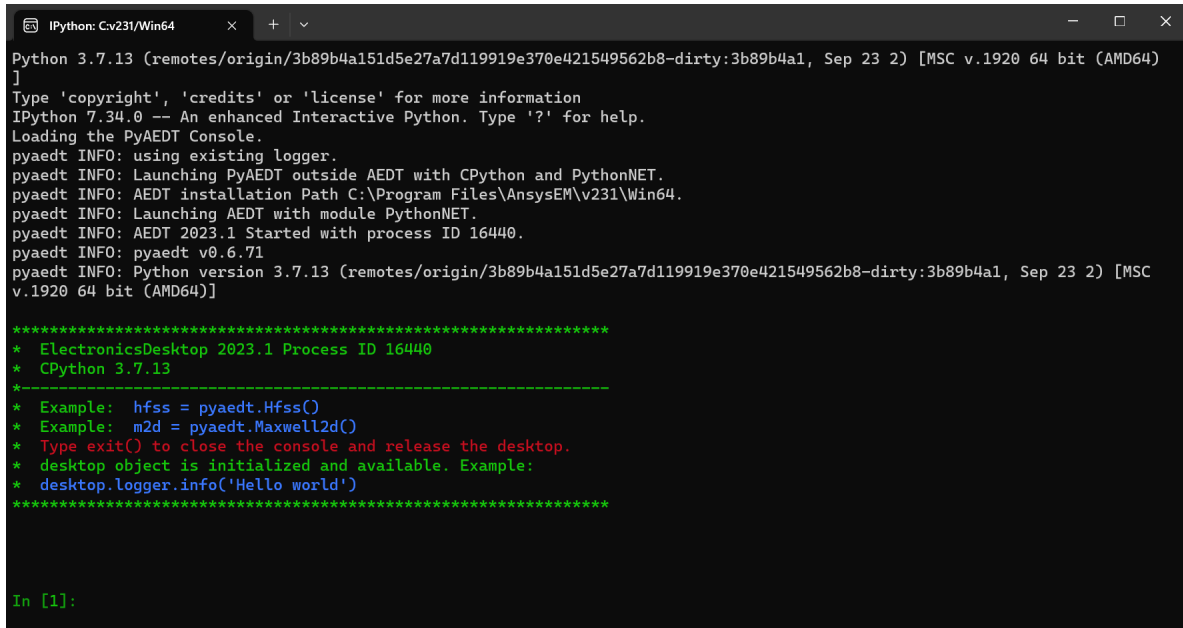
You have multiple options for installing and launching the Magnet Segmentation Toolkit:

- You can install the toolkit directly in AEDT using an installation script and then launch it as a wizard. For more information, see *Install toolkit in AEDT and launch the Magnet Segmentation Toolkit*.
- You can install the toolkit from a Python console and then launch the Magnet Segmentation Toolkit. For more information, see *Install toolkit from Python console and launch the Magnet Segmentation Toolkit*.
- You can install the toolkit from a Python console and then use the toolkits APIs. For more information, see *Install toolkit from Python console and use the toolkits APIs*.

1.2.1 Install toolkit in AEDT and launch the Magnet Segmentation Toolkit

You can install the Magnet Segmentation Toolkit directly in AEDT using the base interpreter from the AEDT installation.

1. From [Install from a Python file](#), follow the steps to install PyAEDT inside AEDT.
2. In AEDT, select **Tools > Toolkit > PyAEDT > Console** to load the PyAEDT console:



```

IPython: C:\v231\Win64
Python 3.7.13 (remotes/origin/3b89b4a151d5e27a7d119919e370e421549562b8-dirty:3b89b4a1, Sep 23 2) [MSC v.1920 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.34.0 -- An enhanced Interactive Python. Type '?' for help.
Loading the PyAEDT Console.
pyaedt INFO: using existing logger.
pyaedt INFO: Launching PyAEDT outside AEDT with CPython and PythonNET.
pyaedt INFO: AEDT installation Path C:\Program Files\AnsysEM\v231\Win64.
pyaedt INFO: Launching AEDT with module PythonNET.
pyaedt INFO: AEDT 2023.1 Started with process ID 16440.
pyaedt INFO: pyaedt v0.6.71
pyaedt INFO: Python version 3.7.13 (remotes/origin/3b89b4a151d5e27a7d119919e370e421549562b8-dirty:3b89b4a1, Sep 23 2) [MSC v.1920 64 bit (AMD64)]

*****
* ElectronicsDesktop 2023.1 Process ID 16440
* CPython 3.7.13
*****
* Example: hfss = pyaedt.Hfss()
* Example: m2d = pyaedt.Maxwell2d()
* Type exit() to close the console and release the desktop.
* desktop object is initialized and available. Example:
* desktop.logger.info('Hello world')
*****

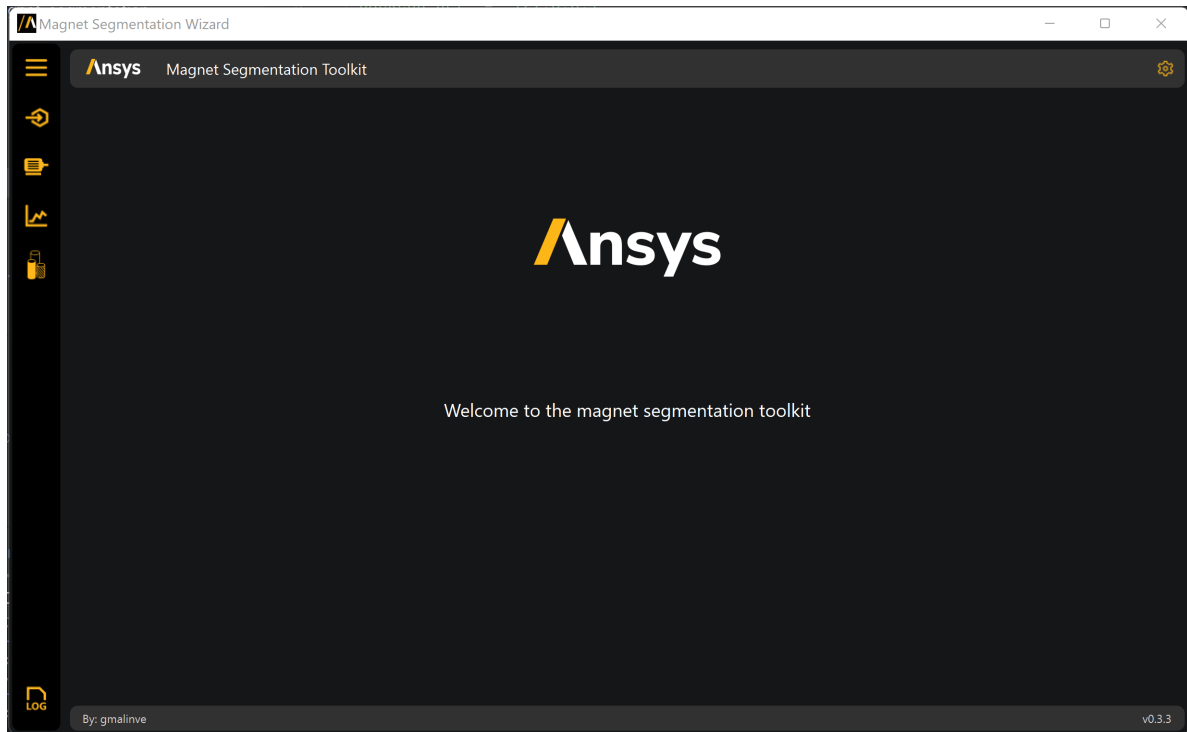
In [1]:

```

3. In the PyAEDT console, run these commands to add the Magnet Segmentation Toolkit as a wizard (toolkit UI) in AEDT:

```
desktop.add_custom_toolkit("MagnetSegmentationWizard")
exit()
```

4. In the AEDT toolbar, click the **MagnetSegmentationWizard** button to open this wizard in AEDT:



The wizard is connected directly to the AEDT session. For wizard usage information, see *UI reference*.

1.2.2 Install toolkit from Python console and launch the Magnet Segmentation Toolkit

You can install the Magnet Segmentation Toolkit in a specific Python environment from the AEDT console.

Note

If you have an existing virtual environment, skip step 1.

Note

If you have already installed the toolkit in your virtual environment, skip step 2.

1. Create a fresh-clean Python environment and activate it:

```
# Create a virtual environment
python -m venv .venv

# Activate it in a POSIX system
source .venv/bin/activate

# Activate it in a Windows CMD environment
.venv\Scripts\activate.bat

# Activate it in Windows PowerShell
.venv\Scripts\Activate.ps1
```

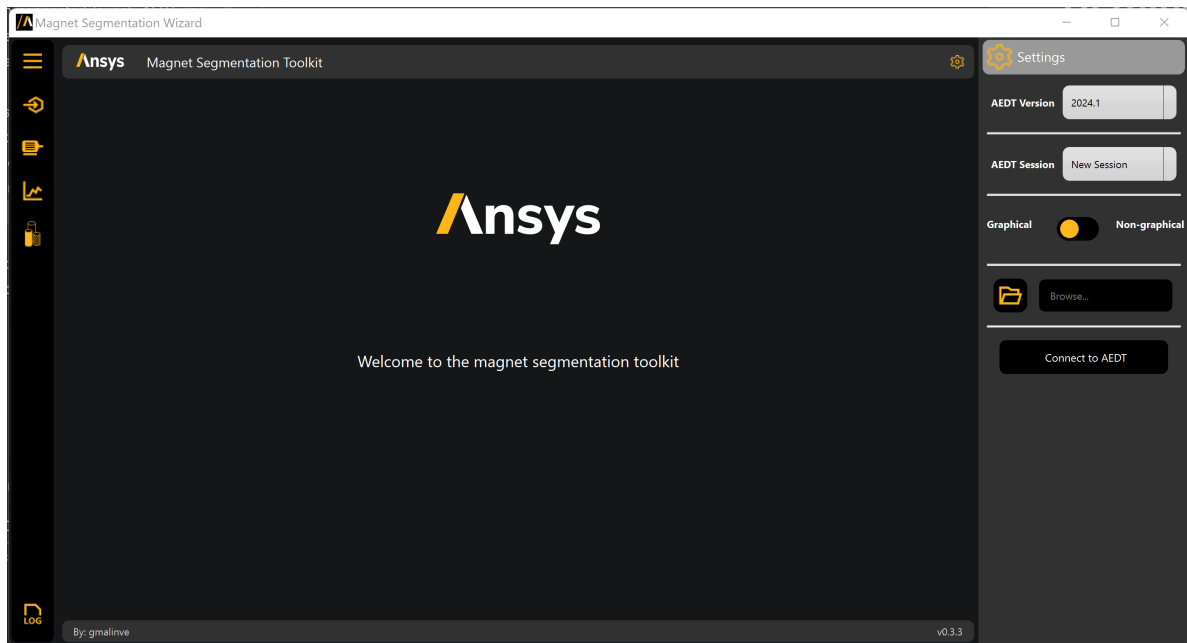

2. Install the toolkit from the GitHub repository:

```
python -m pip install ansys-magnet-segmentation-toolkit
```

3. Launch the Magnet Segmentation Toolkit Wizard:

```
python .venv\Lib\site-packages\ansys\aedt\toolkits\magnet_segmentation\run_toolkit.  
→py
```

4. On the **AEDT Settings** tab, create an AEDT session or connect to an existing one:



For wizard usage information, see *UI reference*.

1.2.3 Install toolkit from Python console and use the toolkits APIs

You can install the toolkit in a specific Python environment and use the toolkits APIs. The code example included in this topic shows how to use the APIs at the model level and toolkit level.

Note

If you have an existing virtual environment, skip step 1.

Note

If you have already installed the toolkit in your virtual environment, skip step 2.

1. Create a fresh-clean Python environment and activate it:

```
# Create a virtual environment  
python -m venv .venv  
  
# Activate it in a POSIX system
```

(continues on next page)

(continued from previous page)

```
source .venv/bin/activate

# Activate it in a Windows CMD environment
.venv\Scripts\activate.bat

# Activate it in Windows PowerShell
.venv\Scripts\Activate.ps1
```

2. Install the toolkit from the GitHub repository:

```
python -m pip install ansys-magnet-segmentation-toolkit
```

3. Open a Python console in your virtual environment:

```
python
```

4. From the command line, use the toolkit to perform segmentation and skew.

Use the toolkit's APIs to import the toolkit, launch AEDT, open a 3D motor model, and then segment and skew this model in Maxwell 3D:

```
# Import backend services
from ansys.aedt.toolkits.magnet_segmentation.backend.api import _
↳ ToolkitBackend

# Backend object
toolkit = ToolkitBackend()

# Get service properties
properties = toolkit.get_properties()

# Define properties

properties["active_project"] = active_project
properties["active_design"] = active_design
properties["design_list"] = {active_project: [active_design]}
properties["is_skewed"] = False
properties["rotor_material"] = "M250-35A_20C"
properties["stator_material"] = "M250-35A_20C"
properties["magnets_material"] = "N30UH_65C"
properties["magnet_segments_per_slice"] = 2
properties["rotor_slices"] = 2
properties["apply_mesh_sheets"] = True
properties["mesh_sheets_number"] = 3

# Set service properties
toolkit.set_properties(properties)

# Launch AEDT, open project and connect to Maxwell3d design
toolkit.launch_aedt()
toolkit.open_project(aedt_file)
toolkit.connect_design("Maxwell3D")
```

(continues on next page)

(continued from previous page)

```
# Segment and skew motor
toolkit.segmentation()
toolkit.apply_skew()

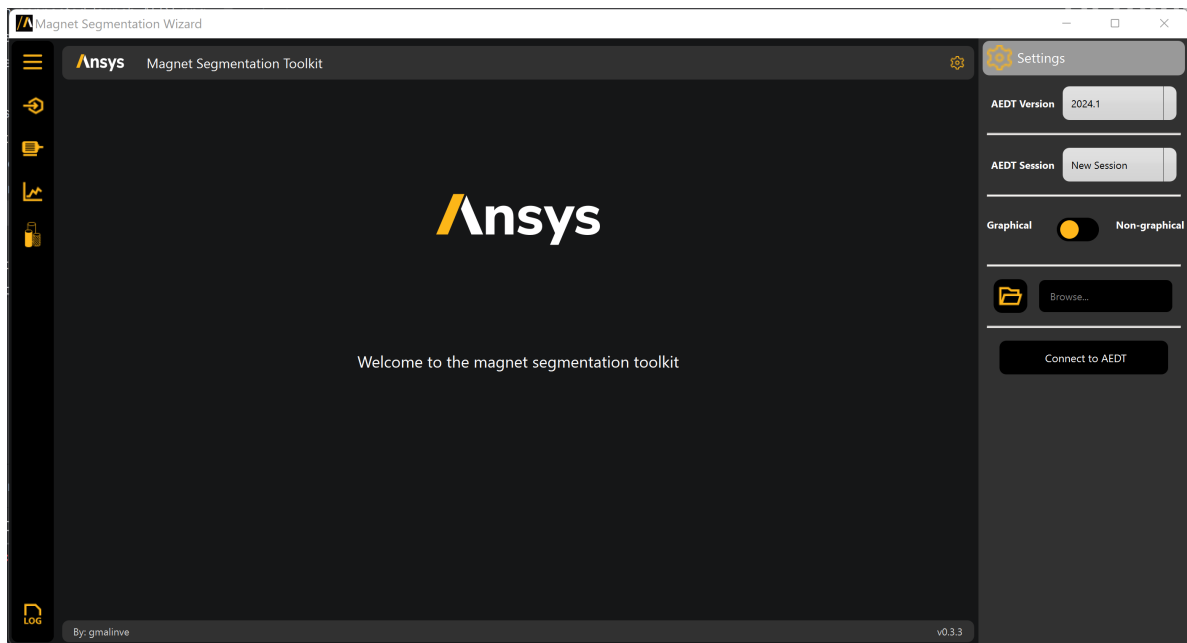
# Release AEDT
service.release_aedt()
```

For descriptions of the APIs available for the Magnet Segmentation Toolkit, see *API reference*.

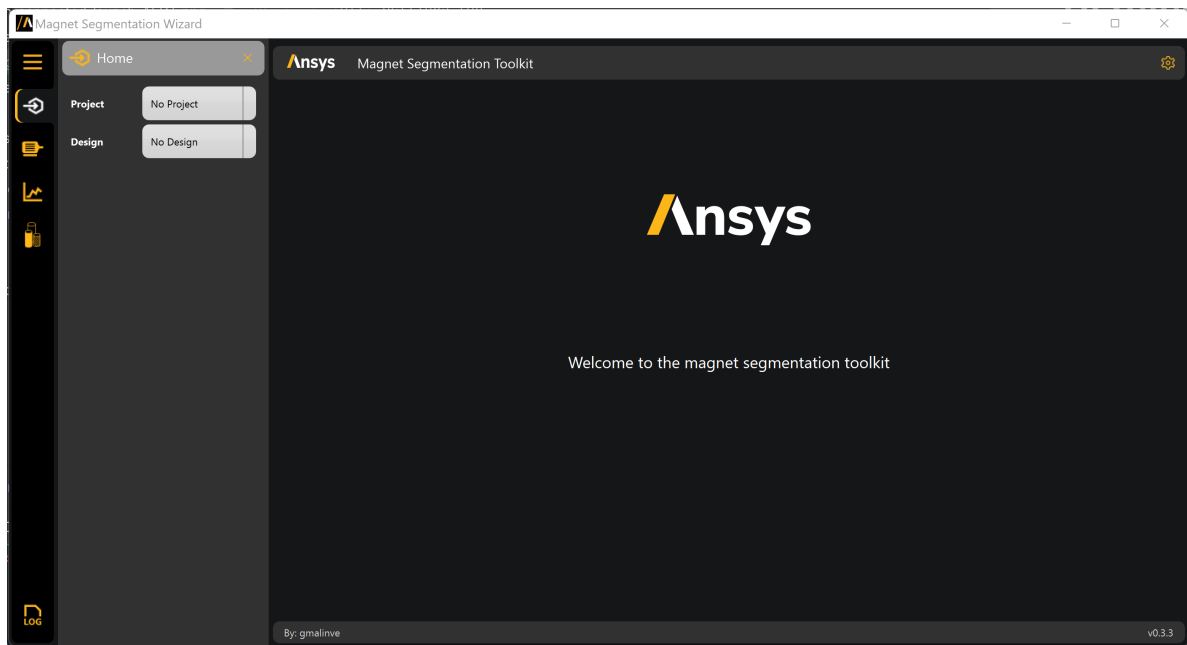
UI REFERENCE

This section describes how to use the Magnet Segmentation Toolkit Wizard, which requires an installed and licensed copy of AEDT. It assumes that you have already launched the wizard from either the AEDT menu or AEDT console. For toolkit installation and wizard launching information, see these topics:

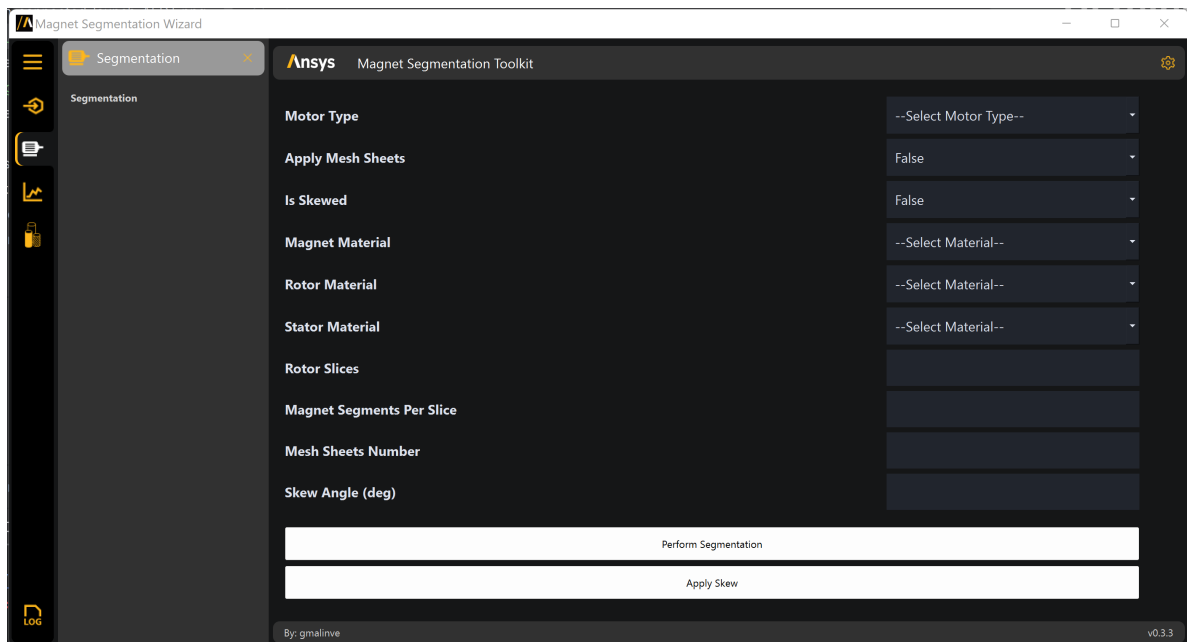
- *Install toolkit in AEDT and launch the Magnet Segmentation Toolkit*
 - *Install toolkit from Python console and launch the Magnet Segmentation Toolkit*
1. On the **Settings** tab, specify settings for either creating an AEDT session or connecting to an existing AEDT session and click **Connect to AEDT**.



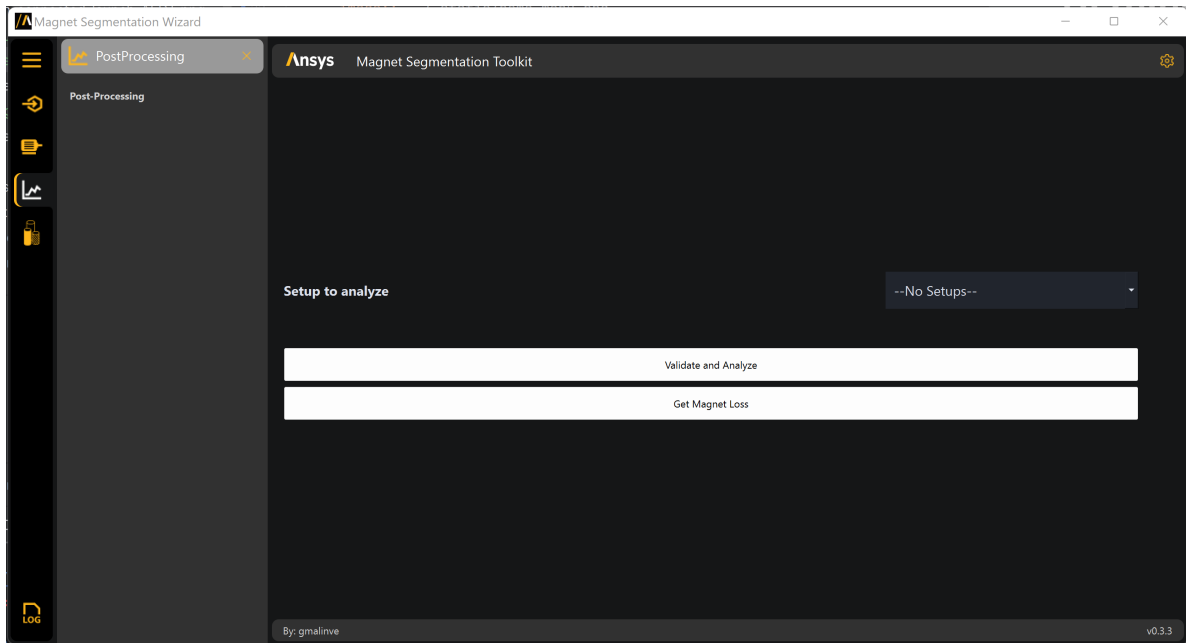
2. Choose project and design from the drop-down list in the **Home menu**.



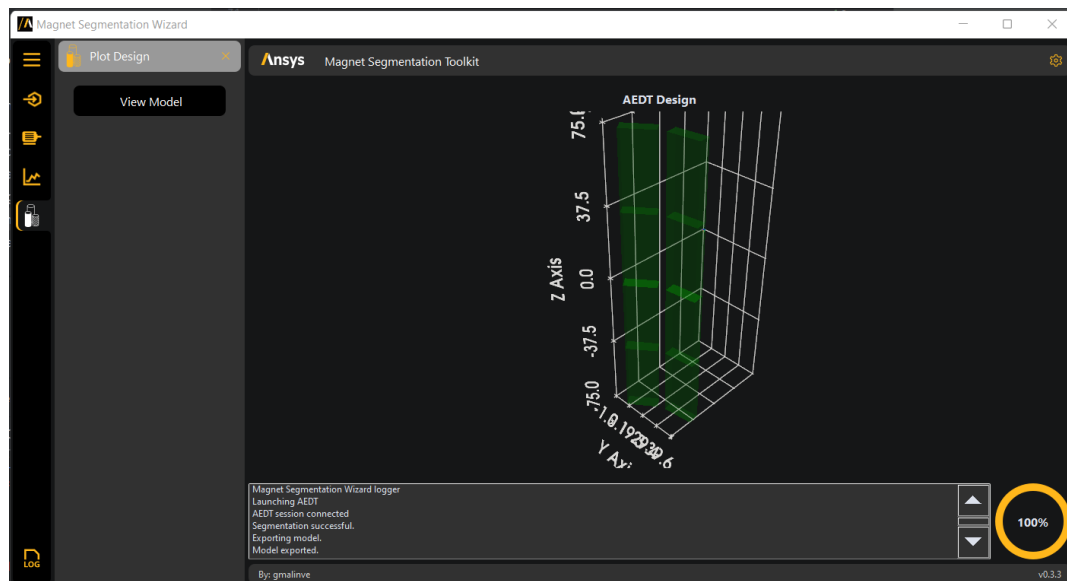
3. Click on the **Segmentation** menu to specify segmentation settings.



4. At the bottom of the tab, click **Perform Segmentation** and then **Apply Skew**.
5. Click on the **Post-processing** menu to select the desired setup to validate and analyze.



6. Click on **Get Magnet Loss** to automatically compute Magnet Loss in AEDT. The report will be automatically generated in AEDT.
7. Click on the **Design** menu to visualize within the toolkit the segmented magnets.



8. The wizard has a progress circle and a logger box where you can see the status of every operation. Every operation must wait for the previous operation to release the toolkit.



API REFERENCE

This section provides descriptions of the available API for the Magnet Segmentation Toolkit:

- **Toolkit API:** Contains the `ToolkitBackend` class, which provides methods for controlling the toolkit workflow. In addition to methods for creating an AEDT session or connecting to an existing AEDT session, this API provides methods for automating the segmentation and skew of a 3D motor.

Warning

Both segmentation and skew of a 3D motor have requirements on the AEDT active project. Ensure that the active design meets these requirements:

- For segmentation, `SymmetryFactor` and `HalfAxial` design settings must be defined.
- For skew, `Shaft` must be the name of the shaft.

3.1 Toolkit API

The Toolkit API contains the `ToolkitBackend` class, which provides methods for controlling the toolkit workflow. In addition to methods for creating an AEDT session or connecting to an existing AEDT session, this API provides methods for automating the segmentation and skew of a 3D motor.

Warning

Both segmentation and skew of a 3D motor have requirements on the AEDT active project. Ensure that the active design meets these requirements:

- For segmentation, `SymmetryFactor` and `HalfAxial` design settings must be defined.
- For skew, `Shaft` must be the name of the shaft.

<code>ToolkitBackend()</code>

Provides methods for controlling the toolkit workflow.
--

3.1.1 `ansys.aedt.toolkits.magnet_segmentation.backend.api.ToolkitBackend`

class `ansys.aedt.toolkits.magnet_segmentation.backend.api.ToolkitBackend`

Provides methods for controlling the toolkit workflow.

This class provides methods for creating an AEDT session, connecting to an existing AEDT session, and automating the segmentation and skew of a 3D motor model.

Examples

```
>>> from ansys.aedt.toolkits.magnet_segmentation.backend.api import ToolkitBackend
>>> toolkit = ToolkitBackend()
>>> toolkit.launch_aedt()
```

`__init__()`

Methods

<code>__init__()</code>	
<code>aedt_sessions()</code>	Get information for the active AEDT sessions.
<code>apply_skew()</code>	Apply skew to rotor slices.
<code>connect_aedt()</code>	Connect to an existing AEDT session.
<code>connect_design([app_name])</code>	Connect to an application design.
<code>export_aedt_model([obj_list, export_path, ...])</code>	Export the model in the OBJ format and then encode the file if the encode parameter is enabled.
<code>get_design_names()</code>	Get the design names for a specific project.
<code>get_magnet_loss()</code>	Get magnet loss.
<code>get_project_name(project_path)</code>	Get the project name from the project path.
<code>get_properties()</code>	Get the toolkit properties.
<code>get_thread_status()</code>	Get the toolkit thread status.
<code>installed_aedt_version()</code>	Get the installed AEDT versions.
<code>is_aedt_connected()</code>	Check if AEDT is connected.
<code>launch_aedt()</code>	Launch AEDT.
<code>launch_thread(process)</code>	Launch the thread.
<code>open_project([project_name])</code>	Open an AEDT project.
<code>release_aedt([close_projects, close_on_exit])</code>	Release AEDT.
<code>save_project([project_path, release_aedt])</code>	Save the project.
<code>segmentation()</code>	Apply object segmentation.
<code>serialize_obj_base64(file_path)</code>	Encode a bytes-like object.
<code>set_properties(data)</code>	Assign the passed data to the internal data model.
<code>validate_and_analyze()</code>	Validate and analyze the design.
<code>wait_to_be_idle([timeout])</code>	Wait for the thread to be idle and ready to accept a new task.

This code shows how to use the ToolkitBackend class:

```
# Import required modules and backend
from ansys.aedt.toolkits.magnet_segmentation.backend.api import ToolkitBackend

# Initialize generic service
toolkit = ToolkitBackend()

# Get the default properties loaded from JSON file
properties = toolkit.get_properties()

# Set properties
aedt_file = os.path.join(common_temp_dir, "input_data", f"{PROJECT_NAME}.aedt")
new_properties = {
    "aedt_version": "2024.1",
```

(continues on next page)

(continued from previous page)

```
"non_graphical": True,
"active_project": aedt_file,
"active_design": DESIGN_NAME,
"design_list": {PROJECT_NAME: [DESIGN_NAME]},
"use_grpc": config["use_grpc"],
}
toolkit.set_properties(new_properties)

# Launch AEDT and open project
msg = toolkit.launch_aedt()
toolkit.open_project(aedt_file)
toolkit.connect_design("Maxwell3D")

# Segment and skew motor
toolkit.segmentation()
toolkit.apply_skew()

# Release AEDT
toolkit.release_aedt()
```


EXAMPLES

End-to-end example show how you can use the Magnet Segmentation Toolkit.

4.1 Maxwell 3D segmentation

This example shows how to use the Magnet Segmentation Toolkit to segment your AEDT motor model.

4.1.1 Perform required imports

Perform required imports.

```
[1]: import os
import shutil
import tempfile

from ansys.aedt.toolkits.common.utils import download_file

from ansys.aedt.toolkits.magnet_segmentation.backend.api import ToolkitBackend
```

4.1.2 Initialize temporary folder and project settings

Initialize a temporary folder to copy the input file into and specify project settings.

```
[2]: URL_BASE = "https://raw.githubusercontent.com/ansys/example-data/master/toolkits/magnet_
↪segmentation/"
AEDT_PROJECT = "e9_eMobility_IPM_3D"
URL = os.path.join(URL_BASE, AEDT_PROJECT + ".aedt")

temp_dir = tempfile.TemporaryDirectory(suffix=".ansys")
active_project = os.path.join(temp_dir.name, AEDT_PROJECT + ".aedt")
download_file(URL, active_project)
active_design = "e9_eMobility_IPM_3D_test"
```

4.1.3 Initialize toolkit

Initialize the toolkit.

```
[3]: toolkit = ToolkitBackend()
```

4.1.4 Get toolkit properties

Get the toolkit properties.

```
[4]: properties = toolkit.get_properties()
```

4.1.5 Initialize properties

Initialize a dictionary of properties.

```
[5]: properties["aedt_version"] = "2025.1"
properties["active_project"] = AEDT_PROJECT
properties["active_design"] = active_design
properties["design_list"] = {AEDT_PROJECT: [active_design]}
properties["is_skewed"] = False
properties["rotor_material"] = "M250-35A_20C"
properties["stator_material"] = "M250-35A_20C"
properties["magnets_material"] = "N30UH_65C"
properties["magnet_segments_per_slice"] = 2
properties["rotor_slices"] = 2
properties["apply_mesh_sheets"] = False
# properties["mesh_sheets_number"] = 3
properties["skew_angle"] = "2deg"
```

4.1.6 Set non-graphical mode

Set non-graphical mode. The default value is False.

```
[6]: properties["non_graphical"] = False
```

4.1.7 Set properties

Set properties.

```
[7]: toolkit.set_properties(properties)

INFO - Updating internal properties.
DEBUG - Updating 'is_skewed' with value False
DEBUG - Updating 'apply_mesh_sheets' with value False
DEBUG - Updating 'magnets_material' with value N30UH_65C
DEBUG - Updating 'rotor_material' with value M250-35A_20C
DEBUG - Updating 'stator_material' with value M250-35A_20C
DEBUG - Updating 'rotor_slices' with value 2
DEBUG - Updating 'magnet_segments_per_slice' with value 2
DEBUG - Updating 'mesh_sheets_number' with value 0
DEBUG - Updating 'skew_angle' with value 2deg
DEBUG - Updating 'setup_to_analyze' with value Setup1
DEBUG - Updating 'objects' with value []
DEBUG - Updating 'aedt_version' with value 2025.1
DEBUG - Updating 'non_graphical' with value False
DEBUG - Updating 'active_project' with value e9_eMobility_IPM_3D
DEBUG - Updating 'active_design' with value e9_eMobility_IPM_3D_test
DEBUG - Updating 'project_list' with value []
DEBUG - Updating 'design_list' with value {'e9_eMobility_IPM_3D': ['e9_eMobility_IPM_3D_
```

(continues on next page)

(continued from previous page)

```

↪test'}}
DEBUG - Updating 'selected_process' with value 0
DEBUG - Updating 'use_grpc' with value True
DEBUG - Updating 'is_toolkit_busy' with value False
DEBUG - Updating 'url' with value 127.0.0.1
DEBUG - Updating 'port' with value 5001
DEBUG - Updating 'debug' with value True
DEBUG - Updating 'toolkit_name' with value common
DEBUG - Updating 'log_file' with value common_backend.log
DEBUG - Properties were updated successfully.

```

[7]: (True, 'Properties were updated successfully.')

4.1.8 Initialize AEDT

Launch a new AEDT session.

[8]: toolkit.launch_aedt()

```

DEBUG - Toolkit is not connected to AEDT.
DEBUG - Launching AEDT.

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC_
↪v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Initializing new Desktop session.
PyAEDT WARNING: Argument `specified_version` is deprecated for method `__init__`; use_
↪`version` instead.
PyAEDT WARNING: Argument `new_desktop_session` is deprecated for method `__init__`; use_
↪`new_desktop` instead.
PyAEDT INFO: Log on console is enabled.
PyAEDT INFO: Log on file C:\Users\ansys\AppData\Local\Temp\pyaedt_ansys_62d6a8e3-2602-
↪4fe7-b824-058ada376295.log is enabled.
PyAEDT INFO: Log on AEDT is disabled.
PyAEDT INFO: Debug logger is disabled. PyAEDT methods will not be logged.
PyAEDT INFO: Launching PyAEDT with gRPC plugin.
PyAEDT INFO: New AEDT session is starting on gRPC port 54451.
PyAEDT INFO: Electronics Desktop started on gRPC port: 54451 after 6.164813756942749_
↪seconds.
PyAEDT INFO: AEDT installation Path C:\Program Files\ANSYS Inc\v251\AnsysEM
PyAEDT INFO: Ansoft.ElectronicsDesktop.2025.1 version started with process ID 2268.

DEBUG - AEDT launched.
DEBUG - Grpc port 54451.

PyAEDT INFO: Desktop has been released.

INFO - AEDT is released.
DEBUG - AEDT is released and project properties are loaded.

```

[8]: True

4.1.9 Open project

Open the project.

```
[9]: toolkit.open_project(active_project)

DEBUG - Toolkit is not connected to AEDT.
DEBUG - Connecting AEDT.

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC_
↪v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Initializing new Desktop session.
PyAEDT WARNING: Argument `specified_version` is deprecated for method `__init__`; use_
↪`version` instead.
PyAEDT WARNING: Argument `new_desktop_session` is deprecated for method `__init__`; use_
↪`new_desktop` instead.
PyAEDT INFO: Log on console is enabled.
PyAEDT INFO: Log on file C:\Users\ansys\AppData\Local\Temp\pyaedt_ansys_62d6a8e3-2602-
↪4fe7-b824-058ada376295.log is enabled.
PyAEDT INFO: Log on AEDT is disabled.
PyAEDT INFO: Debug logger is disabled. PyAEDT methods will not be logged.
PyAEDT INFO: Launching PyAEDT with gRPC plugin.
PyAEDT INFO: Connecting to AEDT session on gRPC port 54451.
PyAEDT INFO: AEDT installation Path C:\Program Files\ANSYS Inc\v251\AnsysEM

DEBUG - Toolkit is connected to AEDT.
DEBUG - Project C:\Users\ansys\AppData\Local\Temp\tmppapz3svs.ansys\e9_eMobility_IPM_3D.
↪aedt is opened
DEBUG - Project name: e9_eMobility_IPM_3D
INFO - Updating internal properties.
DEBUG - Updating 'project_list' with value ['C:/Users/ansys/AppData/Local/Temp/
↪tmppapz3svs.ansys/e9_eMobility_IPM_3D.aedt']
DEBUG - Updating 'active_design' with value e9
DEBUG - Updating 'active_project' with value C:/Users/ansys/AppData/Local/Temp/
↪tmppapz3svs.ansys/e9_eMobility_IPM_3D.aedt
DEBUG - Updating 'design_list' with value {'e9_eMobility_IPM_3D': ['e9']}
DEBUG - Properties were updated successfully.

PyAEDT INFO: Desktop has been released.

INFO - AEDT is released.

[9]: True
```

4.1.10 Connect design

Connect or create a new design.

```
[10]: toolkit.connect_design()

DEBUG - Toolkit is not connected to AEDT.
DEBUG - Connecting AEDT.

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC_
↪v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Initializing new Desktop session.
```

(continues on next page)

(continued from previous page)

```

PyAEDT WARNING: Argument `specified_version` is deprecated for method `__init__`; use
↳ `version` instead.
PyAEDT WARNING: Argument `new_desktop_session` is deprecated for method `__init__`; use
↳ `new_desktop` instead.
PyAEDT INFO: Log on console is enabled.
PyAEDT INFO: Log on file C:\Users\ansys\AppData\Local\Temp\pyaedt_ansys_62d6a8e3-2602-
↳ 4fe7-b824-058ada376295.log is enabled.
PyAEDT INFO: Log on AEDT is disabled.
PyAEDT INFO: Debug logger is disabled. PyAEDT methods will not be logged.
PyAEDT INFO: Launching PyAEDT with gRPC plugin.
PyAEDT INFO: Connecting to AEDT session on gRPC port 54451.
PyAEDT INFO: AEDT installation Path C:\Program Files\ANSYS Inc\v251\AnsysEM

```

```

DEBUG - Toolkit is connected to AEDT.

```

```

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC
↳ v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Returning found Desktop session with PID 2268!
PyAEDT INFO: Project e9_eMobility_IPM_3D set to active.
PyAEDT INFO: Active Design set to e9
PyAEDT INFO: Aedt Objects correctly read

```

```

INFO - Toolkit is connected to AEDT design.

```

```

[10]: True

```

4.1.11 Apply segmentation

Apply segmentation and assign the relative coordinate system.

```

[11]: toolkit.segmentation()

```

```

PyAEDT INFO: Desktop has been released.

```

```

INFO - AEDT is released.

```

```

DEBUG - Toolkit is not connected to AEDT.

```

```

DEBUG - Connecting AEDT.

```

```

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC
↳ v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Initializing new Desktop session.
PyAEDT WARNING: Argument `specified_version` is deprecated for method `__init__`; use
↳ `version` instead.
PyAEDT WARNING: Argument `new_desktop_session` is deprecated for method `__init__`; use
↳ `new_desktop` instead.
PyAEDT INFO: Log on console is enabled.
PyAEDT INFO: Log on file C:\Users\ansys\AppData\Local\Temp\pyaedt_ansys_62d6a8e3-2602-
↳ 4fe7-b824-058ada376295.log is enabled.
PyAEDT INFO: Log on AEDT is disabled.
PyAEDT INFO: Debug logger is disabled. PyAEDT methods will not be logged.
PyAEDT INFO: Launching PyAEDT with gRPC plugin.
PyAEDT INFO: Connecting to AEDT session on gRPC port 54451.
PyAEDT INFO: AEDT installation Path C:\Program Files\ANSYS Inc\v251\AnsysEM

```


DEBUG - Toolkit is connected to AEDT.

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC_↪v.1929 64 bit (AMD64)].

PyAEDT INFO: PyAEDT version 0.15.3.

PyAEDT INFO: Returning found Desktop session with PID 2268!

PyAEDT INFO: Project e9_eMobility_IPM_3D set to active.

PyAEDT INFO: Active Design set to e9

PyAEDT INFO: Aedt Objects correctly read

INFO - Toolkit is connected to AEDT design.

PyAEDT INFO: Modeler class has been initialized! Elapsed time: 0m 6sec

PyAEDT INFO: Parsing design objects. This operation can take time

PyAEDT INFO: Refreshing bodies from Object Info

PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec

PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC_↪v.1929 64 bit (AMD64)].

PyAEDT INFO: PyAEDT version 0.15.3.

PyAEDT INFO: Returning found Desktop session with PID 2268!

PyAEDT INFO: Project e9_eMobility_IPM_3D set to active.

PyAEDT INFO: Active Design set to e9

PyAEDT INFO: Aedt Objects correctly read

PyAEDT INFO: Active Design set to e10

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC_↪v.1929 64 bit (AMD64)].

PyAEDT INFO: PyAEDT version 0.15.3.

PyAEDT INFO: Returning found Desktop session with PID 2268!

PyAEDT INFO: Project e9_eMobility_IPM_3D set to active.

PyAEDT INFO: Active Design set to e9_1

PyAEDT INFO: Aedt Objects correctly read

INFO - Updating internal properties.

DEBUG - Updating 'is_skewed' with value False

DEBUG - Updating 'apply_mesh_sheets' with value False

DEBUG - Updating 'magnets_material' with value N30UH_65C

DEBUG - Updating 'rotor_material' with value M250-35A_20C

DEBUG - Updating 'stator_material' with value M250-35A_20C

DEBUG - Updating 'rotor_slices' with value 2

DEBUG - Updating 'magnet_segments_per_slice' with value 2

DEBUG - Updating 'mesh_sheets_number' with value 0

DEBUG - Updating 'skew_angle' with value 2deg

DEBUG - Updating 'setup_to_analyze' with value Setup1

DEBUG - Updating 'objects' with value []

DEBUG - Updating 'aedt_version' with value 2025.1

DEBUG - Updating 'non_graphical' with value False

DEBUG - Updating 'active_project' with value C:/Users/ansys/AppData/Local/Temp/↪tmppapz3svs.ansys/e9_eMobility_IPM_3D.aedt

DEBUG - Updating 'active_design' with value e9_1

DEBUG - Updating 'project_list' with value ['C:/Users/ansys/AppData/Local/Temp/↪tmppapz3svs.ansys/e9_eMobility_IPM_3D.aedt']

DEBUG - Updating 'design_list' with value {'e9_eMobility_IPM_3D': ['e9', 'e9_1']}

DEBUG - Updating 'selected_process' with value 54451

DEBUG - Updating 'use_grpc' with value True

(continues on next page)

(continued from previous page)

```

DEBUG - Updating 'is_toolkit_busy' with value False
DEBUG - Updating 'url' with value 127.0.0.1
DEBUG - Updating 'port' with value 5001
DEBUG - Updating 'debug' with value True
DEBUG - Updating 'toolkit_name' with value common
DEBUG - Updating 'log_file' with value common_backend.log
DEBUG - Properties were updated successfully.

PyAEDT INFO: Modeler class has been initialized! Elapsed time: 0m 0sec
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Deleted 1 Objects: Rotor_1_ObjectFromFace1_1.
PyAEDT INFO: Deleted 1 objects
PyAEDT INFO: Parsing C:/Users/ansys/AppData/Local/Temp/tmppapz3svs.ansys/e9_eMobility_
↪ IPM_3D.aedt.
PyAEDT INFO: File C:/Users/ansys/AppData/Local/Temp/tmppapz3svs.ansys/e9_eMobility_IPM_
↪ 3D.aedt correctly loaded. Elapsed time: 0m 2sec
PyAEDT INFO: aedt file load time 1.5461289882659912
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Boundary Insulating L1_1Magnet1N1_1_1_segments has been created.
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Boundary Insulating L1_1Magnet2N1_1_1_segments has been created.
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Boundary Insulating L1_1Magnet1N1_1_1_Split1_segments has been created.
PyAEDT INFO: Parsing design objects. This operation can take time

```

(continues on next page)

(continued from previous page)

```

PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Boundary Insulating L1_1Magnet2N1_1_1_Split1_segments has been created.

INFO - Updating internal properties.
DEBUG - Updating 'is_skewed' with value False
DEBUG - Updating 'apply_mesh_sheets' with value False
DEBUG - Updating 'magnets_material' with value N30UH_65C
DEBUG - Updating 'rotor_material' with value M250-35A_20C
DEBUG - Updating 'stator_material' with value M250-35A_20C
DEBUG - Updating 'rotor_slices' with value 2
DEBUG - Updating 'magnet_segments_per_slice' with value 2
DEBUG - Updating 'mesh_sheets_number' with value 0
DEBUG - Updating 'skew_angle' with value 2deg
DEBUG - Updating 'setup_to_analyze' with value Setup1
DEBUG - Updating 'objects' with value ['L1_1Magnet1N1_1_1', 'L1_1Magnet2N1_1_1', 'L1_
↪1Magnet1N1_1_1_Split1', 'L1_1Magnet2N1_1_1_Split1', 'L1_1Magnet1N1_1_1_ObjectFromFace1_
↪1_ObjectFromFace1', 'L1_1Magnet2N1_1_1_ObjectFromFace1_1_ObjectFromFace1', 'L1_
↪1Magnet1N1_1_1_Split1_ObjectFromFace1_1_ObjectFromFace1', 'L1_1Magnet2N1_1_1_Split1_
↪ObjectFromFace1_1_ObjectFromFace1']
DEBUG - Updating 'aedt_version' with value 2025.1
DEBUG - Updating 'non_graphical' with value False
DEBUG - Updating 'active_project' with value C:/Users/ansys/AppData/Local/Temp/
↪tmppapz3svs.ansys/e9_eMobility_IPM_3D.aedt
DEBUG - Updating 'active_design' with value e9_1
DEBUG - Updating 'project_list' with value ['C:/Users/ansys/AppData/Local/Temp/
↪tmppapz3svs.ansys/e9_eMobility_IPM_3D.aedt']
DEBUG - Updating 'design_list' with value {'e9_eMobility_IPM_3D': ['e9', 'e9_1']}
DEBUG - Updating 'selected_process' with value 54451
DEBUG - Updating 'use_grpc' with value True
DEBUG - Updating 'is_toolkit_busy' with value False
DEBUG - Updating 'url' with value 127.0.0.1
DEBUG - Updating 'port' with value 5001
DEBUG - Updating 'debug' with value True
DEBUG - Updating 'toolkit_name' with value common
DEBUG - Updating 'log_file' with value common_backend.log
DEBUG - Properties were updated successfully.

PyAEDT INFO: Project e9_eMobility_IPM_3D Saved correctly
PyAEDT INFO: Desktop has been released.

INFO - AEDT is released.

```

[11]: True

4.1.12 Apply skew angle

Apply the skew angle to rotor slices.

```
[12]: toolkit.apply_skew()

DEBUG - Toolkit is not connected to AEDT.
DEBUG - Connecting AEDT.

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC_
↪v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Initializing new Desktop session.
PyAEDT WARNING: Argument `specified_version` is deprecated for method `__init__`; use_
↪`version` instead.
PyAEDT WARNING: Argument `new_desktop_session` is deprecated for method `__init__`; use_
↪`new_desktop` instead.
PyAEDT INFO: Log on console is enabled.
PyAEDT INFO: Log on file C:\Users\ansys\AppData\Local\Temp\pyaedt_ansys_62d6a8e3-2602-
↪4fe7-b824-058ada376295.log is enabled.
PyAEDT INFO: Log on AEDT is disabled.
PyAEDT INFO: Debug logger is disabled. PyAEDT methods will not be logged.
PyAEDT INFO: Launching PyAEDT with gRPC plugin.
PyAEDT INFO: Connecting to AEDT session on gRPC port 54451.
PyAEDT INFO: AEDT installation Path C:\Program Files\ANSYS Inc\v251\AnsysEM

DEBUG - Toolkit is connected to AEDT.

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC_
↪v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Returning found Desktop session with PID 2268!
PyAEDT INFO: Project e9_eMobility_IPM_3D set to active.
PyAEDT INFO: Active Design set to e9_1
PyAEDT INFO: Aedt Objects correctly read

INFO - Toolkit is connected to AEDT design.

PyAEDT INFO: Modeler class has been initialized! Elapsed time: 0m 0sec
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Parsing C:/Users/ansys/AppData/Local/Temp/tmppapz3svs.ansys/e9_eMobility_
↪IPM_3D.aedt.
PyAEDT INFO: File C:/Users/ansys/AppData/Local/Temp/tmppapz3svs.ansys/e9_eMobility_IPM_
↪3D.aedt correctly loaded. Elapsed time: 0m 1sec
PyAEDT INFO: aedt file load time 1.4508190155029297
PyAEDT INFO: Parsing design objects. This operation can take time
PyAEDT INFO: Refreshing bodies from Object Info
PyAEDT INFO: Bodies Info Refreshed Elapsed time: 0m 0sec
PyAEDT INFO: 3D Modeler objects parsed. Elapsed time: 0m 0sec
PyAEDT INFO: Union of 2 objects has been executed.
PyAEDT INFO: Desktop has been released.

INFO - AEDT is released.
```

```
[12]: True
```

4.1.13 Validate and analyze design

Uncomment the line to validate and analyze the design.

```
toolkit.validate_and_analyze()
```

4.1.14 Create magnet loss report

Uncomment the lines to create magnet loss report and compute average value.

```
magnet_loss = toolkit.get_magnet_loss() print(fAverage magnet loss: {magnet_loss} W)
```

4.1.15 Save and release AEDT

Save and release AEDT.

```
toolkit.save_project()
```

```
[13]: toolkit.release_aedt(True, True)

DEBUG - Toolkit is not connected to AEDT.
DEBUG - Connecting AEDT.

PyAEDT INFO: Python version 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC
→v.1929 64 bit (AMD64)].
PyAEDT INFO: PyAEDT version 0.15.3.
PyAEDT INFO: Initializing new Desktop session.
PyAEDT WARNING: Argument `specified_version` is deprecated for method `__init__`; use
→`version` instead.
PyAEDT WARNING: Argument `new_desktop_session` is deprecated for method `__init__`; use
→`new_desktop` instead.
PyAEDT INFO: Log on console is enabled.
PyAEDT INFO: Log on file C:\Users\ansys\AppData\Local\Temp\pyaedt_ansys_62d6a8e3-2602-
→4fe7-b824-058ada376295.log is enabled.
PyAEDT INFO: Log on AEDT is disabled.
PyAEDT INFO: Debug logger is disabled. PyAEDT methods will not be logged.
PyAEDT INFO: Launching PyAEDT with gRPC plugin.
PyAEDT INFO: Connecting to AEDT session on gRPC port 54451.
PyAEDT INFO: AEDT installation Path C:\Program Files\ANSYS Inc\v251\AnsysEM

DEBUG - Toolkit is connected to AEDT.

PyAEDT INFO: Desktop has been released and closed.

INFO - AEDT is released.
```

```
[13]: True
```

4.1.16 Remove temporary folder

Remove the temporary folder.

```
[14]: shutil.rmtree(temp_dir.name, ignore_errors=True)
```

CONTRIBUTE

Overall guidance on contributing to a PyAnsys repository appears in [Contributing](#) in the *PyAnsys developers guide*. Ensure that you are thoroughly familiar with this guide before attempting to contribute to PyAEDT or its toolkits.

The following contribution information is specific to the Magnet Segmentation Toolkit.

5.1 Clone the repository

To clone and install the latest version of the Magnet Segmentation Toolkit in development mode, run these commands:

```
git clone https://github.com/ansys/magnet-segmentation-toolkit.git
cd magnet-segmentation-toolkit
python -m pip install --upgrade pip
pip install -e .
```

5.2 Post issues

Use the [Magnet Segmentation Toolkit Issues](#) page to create issues to report bugs and request new features.

5.3 View documentation

Documentation for the latest stable release is hosted at [Magnet Segmentation Toolkit Documentation](#).

In the upper right corner of the documentations title bar, there is an option for switching from viewing the documentation for the latest stable release to viewing the documentation for the development version or previously released versions.

5.4 Adhere to code style

The Magnet Segmentation Toolkit is compliant with [PyAnsys code style](#). It uses the tool [pre-commit](#) to check the code style.

You can install and activate this tool with these commands:

```
pip install pre-commit
pre-commit run --all-files
```

You can also install this as a pre-commit hook with this command:

```
pre-commit install
```

This way, its not possible for you to push code that fails the style checks:

```
$ pre-commit install
$ git commit -am "Add my cool feature."
black.....Passed
isort (python).....Passed
flake8.....Passed
codespell.....Passed
fix requirements.txt.....Passed
blacken-docs.....Passed
```

5.4.1 Maximum line length

Best practice is to keep the line length at or below 120 characters for code and comments. Lines longer than this might not display properly on some terminals and tools or might be difficult to follow.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`