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

Jul 17, 2025

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

CHAPTER

ONE

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

1.1.1 Download links

The following installers are available for different operating systems:

Table 1: Available Installers

Installer Link	Operating System
Download	Windows
Download	Ubuntu 22.04
Download	Ubuntu 24.04

Visit the Releases page and pull down the latest installer.

1.1.2 Installing the `Magnet Segmentation Toolkit`

Windows

First step is installing the `Magnet Segmentation Toolkit`. In order to do so, follow the next steps.

1. Download the necessary installer from the latest available release. The file should be named `Magnet-Segmentation-Toolkit-Installer.exe`.
2. Execute the installer.
3. Search for the `Magnet Segmentation Toolkit` and run it.

The `Magnet Segmentation Toolkit` window should appear at this stage.

Linux

Ubuntu

Prerequisites:

1. **OS** supported for **Ubuntu(24.04 and 22.04)**.

2. Update `apt-get` repository and install the following packages with **sudo** privileges: **wget, gnome, libffi-dev, libssl-dev, libsqlite3-dev, libxcb-xinerama0 and build-essential** packages with **sudo** privileges

```
sudo apt-get update -y
sudo apt-get install wget gnome libffi-dev libssl-dev libsqlite3-dev libxcb-
↪xinerama0 build-essential -y
```

3. Install **zlib** package

```
wget https://zlib.net/current/zlib.tar.gz
tar xvzf zlib.tar.gz
cd zlib-*
make clean
./configure
make
sudo make install
```

To install the `Magnet Segmentation Toolkit`, follow below steps.

1. Download the necessary installer from the latest available release. The file should be named `Magnet-Segmentation-Toolkit-Installer-ubuntu_*.zip`.
2. Execute the below command on the terminal

```
unzip Magnet-Segmentation-Toolkit-Installer-ubuntu_*.zip
./installer.sh
```

3. Search for the `Magnet Segmentation Toolkit` and run it.

The `Magnet Segmentation Toolkit` window should appear at this stage.

To uninstall the `Magnet Segmentation Toolkit`, follow below steps.

1. Go to File menu. Click Uninstall option.
2. Click `Uninstall` button.

1.1.3 Python Installation

The Magnet Segmentation Toolkit can be installed like any other open source package. From PyPI, you can either install both the backend and user interface (UI) methods or install only the backend methods.

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
```

1.1.4 For developers

You can be up and running with four lines of code:

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

Now you can run it with:

```
run_toolkit
```

Details

Installing Pytools installer in developer mode allows you to modify the source and enhance it.

Before contributing to the project, please refer to the PyAnsys Developers guide. You need to follow these steps:

1. Start by cloning this repository:

   ```
   git clone https://github.com/ansys/magnet-segmentation-toolkit
   ```

2. Create a fresh-clean Python environment and activate it. Refer to the official venv documentation if you require further information:

   ```
   # Create a virtual environment
   python -m venv .venv
   # Activate it in a POSIX system
   source .venv/bin/activate
   # Activate it in Windows CMD environment
   .venv\Scripts\activate.bat
   # Activate it in Windows Powershell
   .venv\Scripts\Activate.ps1
   ```

3. Install the project in editable mode:

   ```
   python -m pip install -e .[tests,doc]
   ```

4. Finally, verify your development installation by running:

   ```
   pytest tests -v
   ```

Style and testing

This project uses pre-commit. Install with:

```
pip install pre-commit
run pre-commit install
```

This now runs `pre-commit` for each commit to ensure you follow project style guidelines. For example:

```
git commit -am 'fix style'
isort....................................................................Passed
black....................................................................Passed
blacken-docs.............................................................Passed
flake8...................................................................Passed
codespell................................................................Passed
pydocstyle...............................................................Passed
check for merge conflicts................................................Passed
debug statements (python)................................................Passed
check yaml...............................................................Passed
trim trailing whitespace.................................................Passed
Validate GitHub Workflows................................................Passed
```

If you need to run it again on all files and not just staged files, run:

```
run pre-commit run --all-files
```

Local build

This app can be deployed as a frozen app using pyinstaller with:

```
pip install -e .[freeze]
run pyinstaller frozen.spec
```

This generates app files at `dist/magnet_segmentation_toolkit` and you can run it locally by executing `Magnet Segmentation Toolkit.exe`. For more information on how to create a standalone executable, refer to this section: How to distribute.

Documentation

For building documentation, you can either run the usual rules provided in the Sphinx Makefile:

```
pip install -e .[doc]
doc/make.bat html
# subsequently open the documentation with (under Linux):
<your_browser_name> doc/html/index.html
```

1.2 User guide

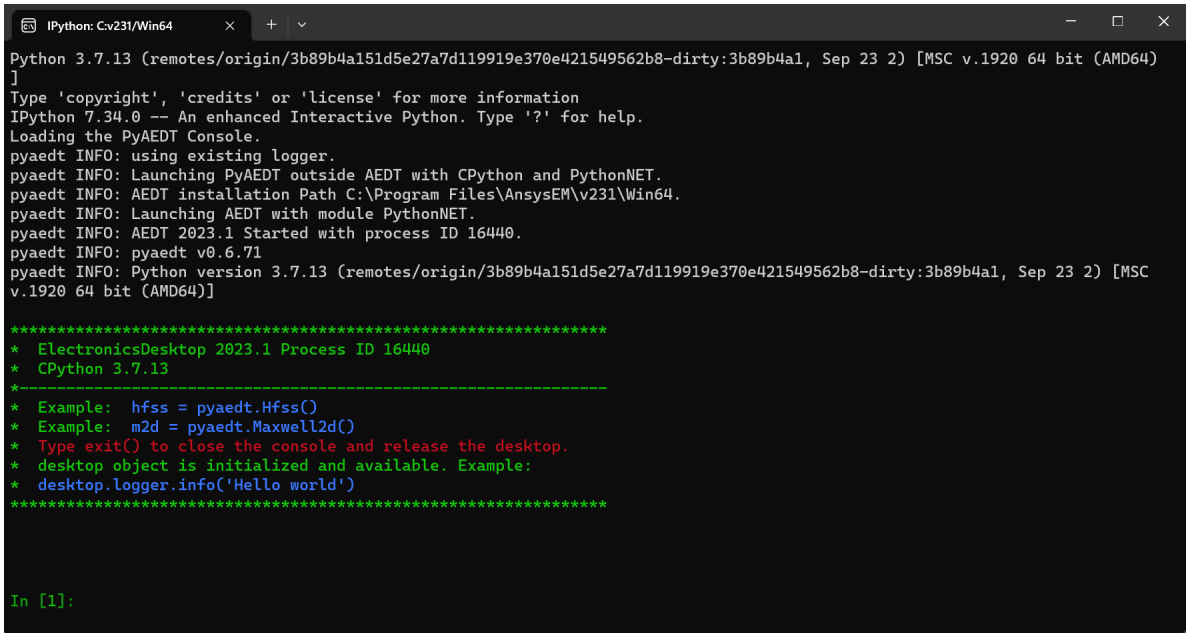
You have multiple options for installing and launching the Magnet Segmentation Toolkit:

- 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.

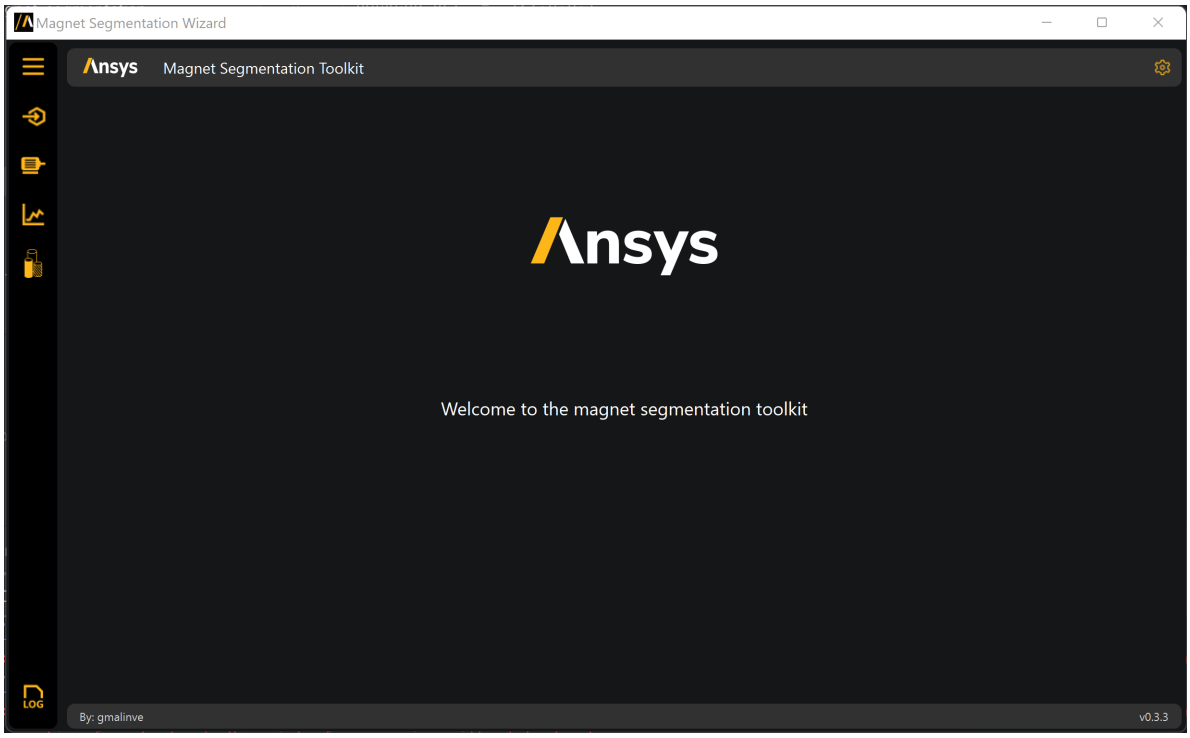
1. From Install from a Python file, follow the steps to install PyAEDT inside AEDT.
2. In AEDT, select **Tools > Toolkit > PyAEDT > Console** to load the PyAEDT console:



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
```

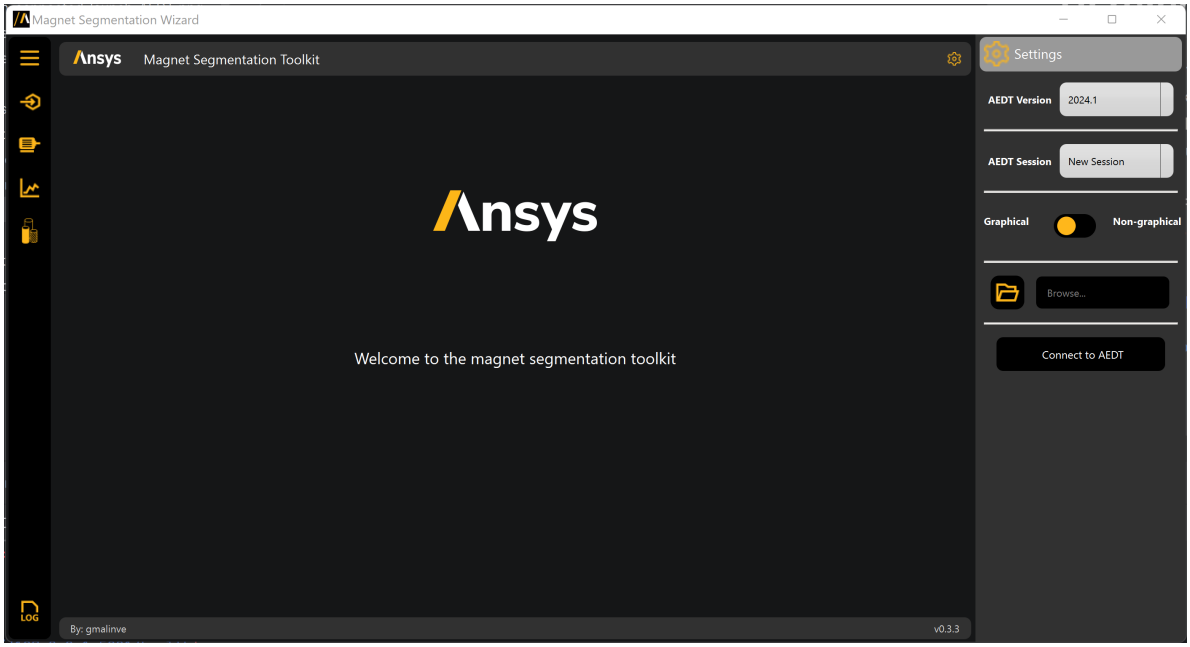
2. Install the toolkit from the GitHub repository:

```
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
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 toolkits 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")

# 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*.

CHAPTER

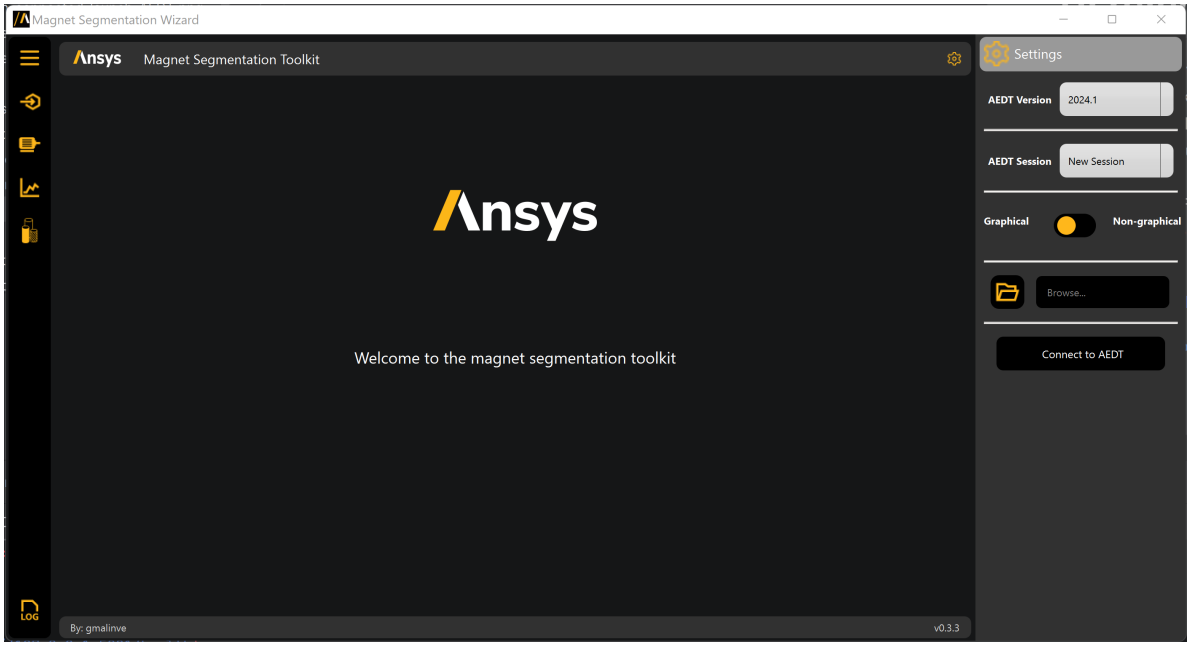
TWO

UI REFERENCE

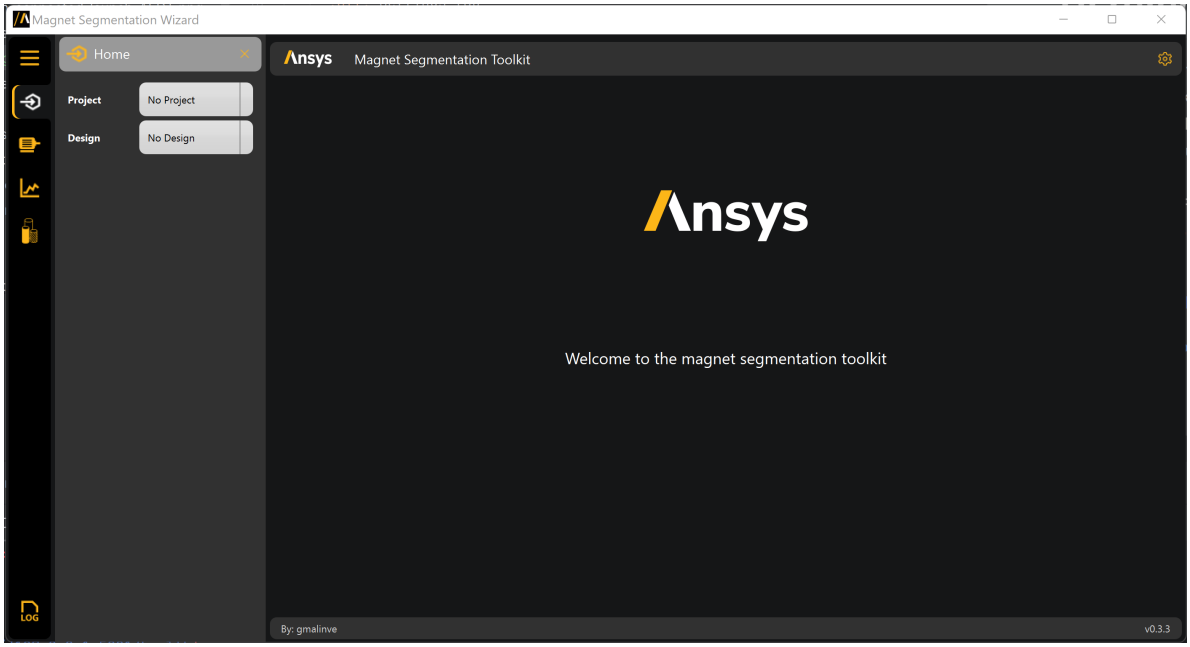
This section describes how to use the Magnet Segmentation Toolkit Wizard, which requires an installed and licensed copy of AEDT. It assumes that you have already launched the wizard from either the AEDT menu or AEDT console. For toolkit installation and wizard launching information, see these topics:

- *Install toolkit in AEDT and launch the Magnet Segmentation Toolkit*
- *Install toolkit from Python console and launch the Magnet Segmentation Toolkit*

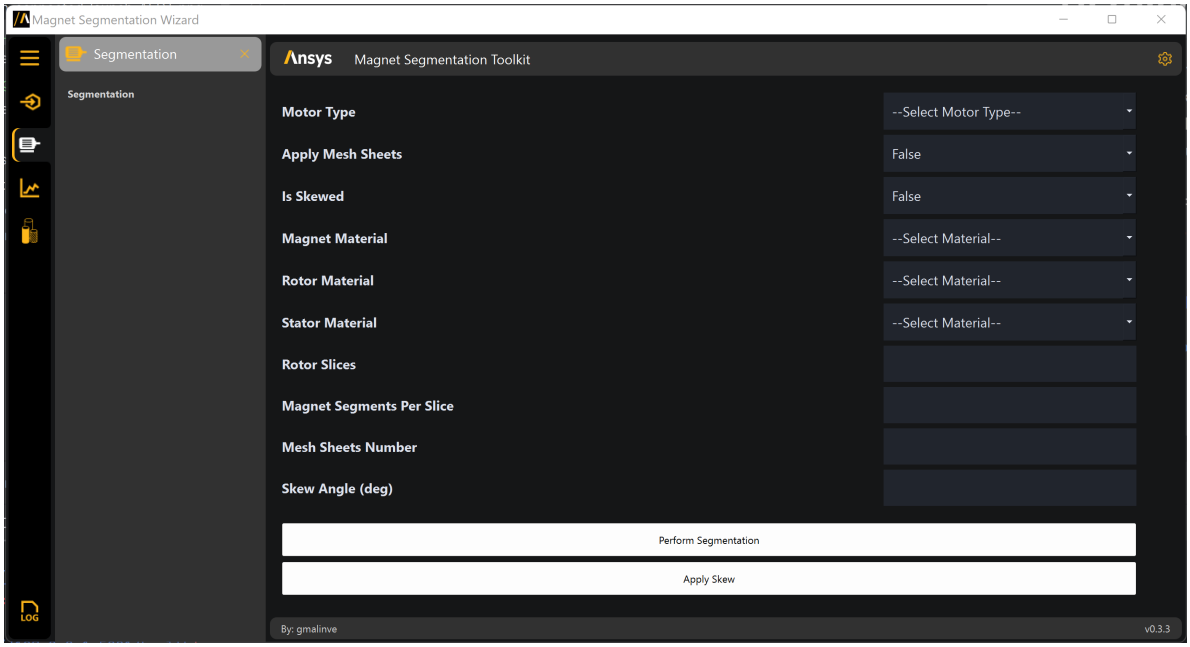
1. On the **Settings** tab, specify settings for either creating an AEDT session or connecting to an existing AEDT session and click **Connect to AEDT**.



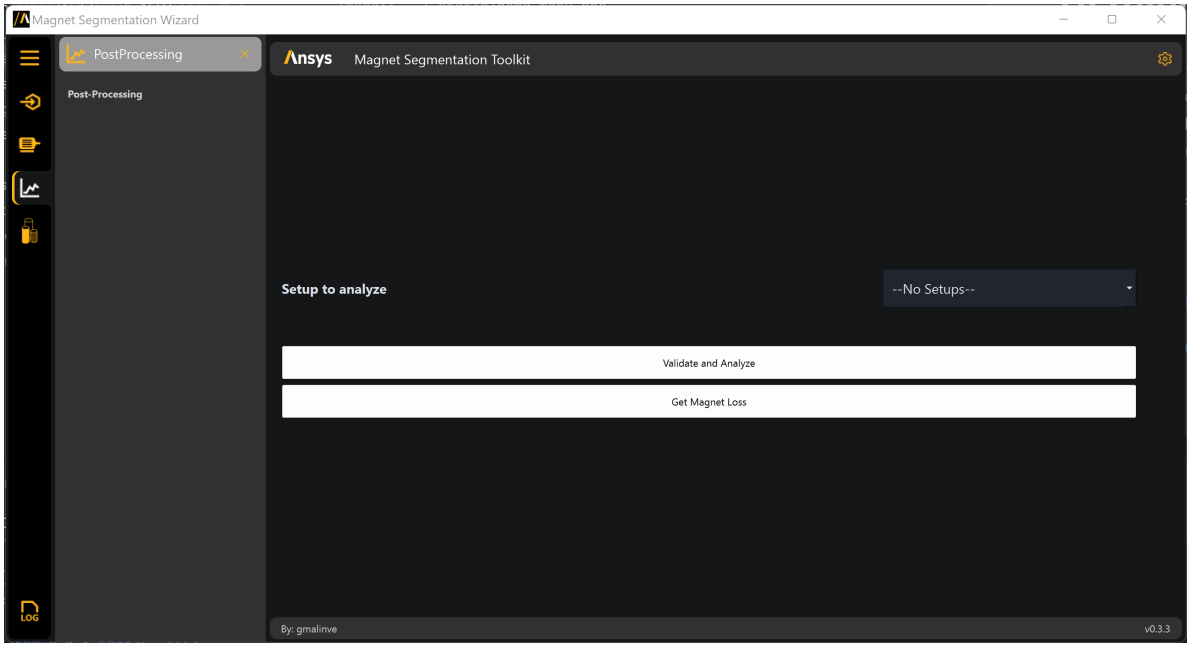
2. Choose project and design from the drop-down list in the **Home menu**.



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

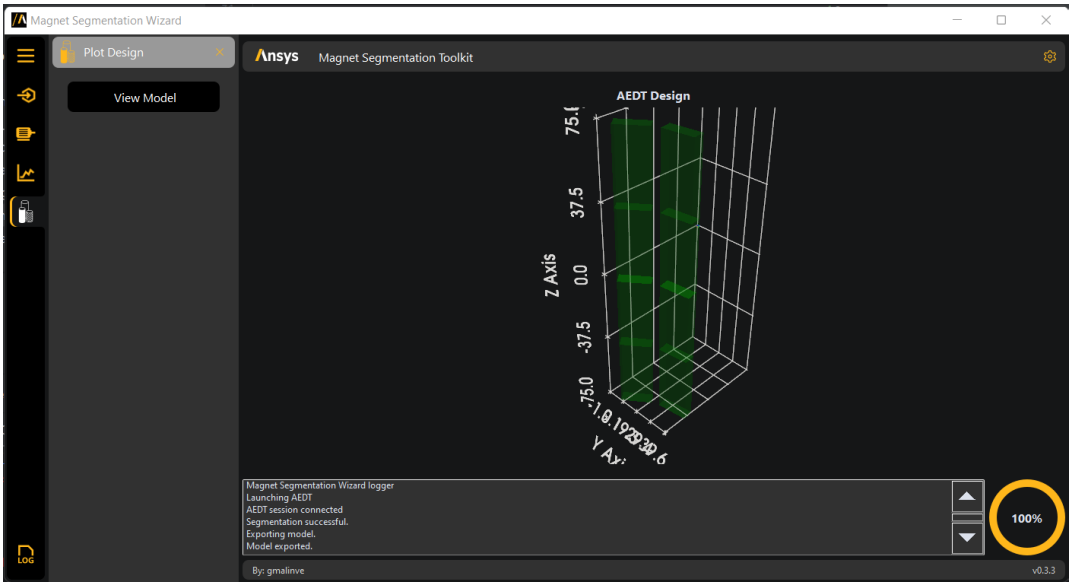


4. At the bottom of the tab, click **Perform Segmentation** and then **Apply Skew**.
5. Click on the **Post-processing menu** to select the desired setup to validate and analyze.

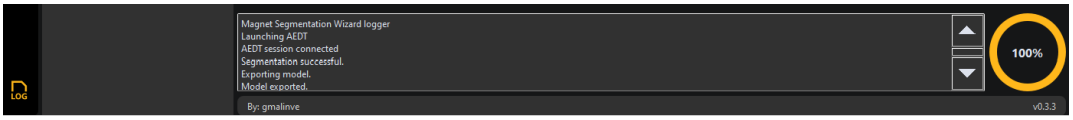


6. Click on **Get Magnet Loss** to automatically compute Magnet Loss in AEDT. The report is automatically generated in AEDT.

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.



CHAPTER

THREE

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.

ToolkitBackend()	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

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

CHAPTER

FOUR

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

4.1.8 Initialize AEDT

Launch a new AEDT session.

```
[8]: toolkit.launch_aedt()
     INFO - AEDT is released.
[8]: True
```

4.1.9 Open project

Open the project.

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

```
INFO - Updating internal properties.
INFO - AEDT is released.
```

[9]:
```
True
```

4.1.10 Connect design

Connect or create a new design.

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

```
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()
```

```
INFO - AEDT is released.
INFO - Toolkit is connected to AEDT design.
INFO - Updating internal properties.
INFO - Updating internal properties.
INFO - AEDT is released.
```

[11]:
```
True
```

4.1.12 Apply skew angle

Apply the skew angle to rotor slices.

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

```
INFO - Toolkit is connected to AEDT design.
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)
```

```
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)
```

CHAPTER

FIVE

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.

CHAPTER

SIX

RELEASE NOTES

This document contains the release notes for the project.

6.1 0.8.5 - July 17, 2025

Fixed

Fix download links #274

Maintenance

Update changelog for v0.8.3 #273

6.2 0.8.4 - July 17, 2025

No significant changes. 0.8.3 - July 10, 2025 ==========

Dependencies

Bump actions/download-artifact from 4.1.9 to 4.3.0 #259

Fixed

Ci cd #271

Maintenance

Update changelog for v0.8.2 #269

6.3 0.8.2 - July 08, 2025

Fixed

Populate props #268

Maintenance

Update changelog for v0.8.1	#265

6.4 0.8.1 - June 13, 2025

Dependencies

Bump ansys/actions from 9 to 10	#262

Documentation

update installation.rst	#263

Maintenance

update CHANGELOG for v0.8.0	#260
update - 0.9.dev0	#261

6.5 0.8.0 - June 03, 2025

Fixed

improve skew	#251
rotate insulation sheet + update cicd pypi	#258

Maintenance

update CHANGELOG for v0.7.0	#249
Update v0.8.dev0	#250
update cicd and add linux tests	#254
add codecov.yml	#256

6.6 0.7.0 - May 08, 2025

Documentation

Update `CONTRIBUTORS.md` with the latest contributors	#243

Fixed

Fix release artifacts #248

Miscellaneous

Create installer #245

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search