FOSSEE Semester Long Internship Report

On

Development of Offline Winodws installer for Osdag

Submitted by

Mehendi Hasan

3rd Year B.Sc(H) Physics Major, Computer Science Minor

Kirori Mal College

University of Delhi

Under the Guidance of

Prof. Siddhartha Ghosh

Department of Civil Engineering

Indian Institute of Technology Bombay

Mentors:

Ajmal Babu M S

Parth Karia

Ajinkya Dahale

June 14, 2025

Acknowledgments

- Start with a general statement of thanks. Express your overall gratitude to everyone who supported you during your project or research.

- Project staff at the Osdag team, Ajmal Babu M. S., Ajinkya Dahale, and Parth Karia,

- Osdag Principal Investigator (PI) Prof. Siddhartha Ghosh, Department of Civil Engineering at IIT Bombay

- FOSSEE PI Prof. Kannan M. Moudgalya, FOSSEE Project Investigator, Department of Chemical Engineering, IIT Bombay

- FOSSEE Managers Usha Viswanathan and Vineeta Parmar and their entire team

- Acknowledge the support from the National Mission on Education through Information and Communication Technology (ICT), Ministry of Education (MoE), Government of India, for their role in facilitating this project

- Acknowledge your colleagues who worked with you during your internship or project.

- If appropriate, thank your college, department, head, and principal for their support during your studies.

Contents

Chapter 1

Introduction

1.1 National Mission in Education through ICT

The National Mission on Education through ICT (NMEICT) is a scheme under the Department of Higher Education, Ministry of Education, Government of India. It aims to leverage the potential of ICT to enhance teaching and learning in Higher Education Institutions in an anytime-anywhere mode.

The mission aligns with the three cardinal principles of the Education Policy—**access, equity, and quality**—by:

- Providing connectivity and affordable access devices for learners and institutions.
- Generating high-quality e-content free of cost.

NMEICT seeks to bridge the digital divide by empowering learners and teachers in urban and rural areas, fostering inclusivity in the knowledge economy. Key focus areas include:

- Development of e-learning pedagogies and virtual laboratories.
- Online testing, certification, and mentorship through accessible platforms like EduSAT and DTH.
- Training and empowering teachers to adopt ICT-based teaching methods.

For further details, visit the official website: www.nmeict.ac.in.

1.1.1 ICT Initiatives of MoE

The Ministry of Education (MoE) has launched several ICT initiatives aimed at students, researchers, and institutions. The table below summarizes the key details:

No.	Resource	For Students/Researchers	For Institutions
		Audio-Video e-content	
1	SWAYAM	Earn credit via online courses	Develop and host courses; accept credits
2	SWAYAMPRABHA	Access 24x7 TV programs	Enable SWAYAMPRABHA viewing facilities
		Digital Content Access	
3	National Digital Library	Access e-content in multiple disciplines	List e-content; form NDL Clubs
4	e-PG Pathshala	Access free books and e-content	Host e-books
5	Shodhganga	Access Indian research theses	List institutional theses
6	e-ShodhSindhu	Access full-text e-resources	Access e-resources for institutions
		Hands-on Learning	
7	e-Yantra	Hands-on embedded systems training	Create e-Yantra labs with IIT Bombay
8	FOSSEE	Volunteer for open-source software	Run labs with open-source software
9	Spoken Tutorial	Learn IT skills via tutorials	Provide self-learning IT content
10	Virtual Labs	Perform online experiments	Develop curriculum-based experiments
		E-Governance	
11	SAMARTH ERP	Manage student lifecycle digitally	Enable institutional e-governance
		Tracking and Research Tools	
12	VIDWAN	Register and access experts	Monitor faculty research outcomes
13	Shodh Shuddhi	Ensure plagiarism-free work	Improve research quality and reputation
14	Academic Bank of Credits	Store and transfer credits	Facilitate credit redemption

Table 1.1: Summary of ICT Initiatives by the Ministry of Education

1.2 FOSSEE Project

The FOSSEE (Free/Libre and Open Source Software for Education) project promotes the use of FLOSS tools in academia and research. It is part of the National Mission on Education through Information and Communication Technology (NMEICT), Ministry of Education (MoE), Government of India.

1.2.1 Projects and Activities

The FOSSEE Project supports the use of various FLOSS tools to enhance education and research. Key activities include:

- **Textbook Companion**: Porting solved examples from textbooks using FLOSS.
- **Lab Migration**: Facilitating the migration of proprietary labs to FLOSS alternatives.
- **Niche Software Activities**: Specialized activities to promote niche software tools.
- **Forums**: Providing a collaborative space for users.
- **Workshops and Conferences**: Organizing events to train and inform users.

1.2.2 Fellowships

FOSSEE offers various internship and fellowship opportunities for students:

- Winter Internship
- Summer Fellowship
- Semester-Long Internship

Students from any degree and academic stage can apply for these internships. Selection is based on the completion of screening tasks involving programming, scientific computing, or data collection that benefit the FLOSS community. These tasks are designed to be completed within a week.

For more details, visit the official FOSSEE website.

Figure 1.1: FOSSEE Projects and Activities

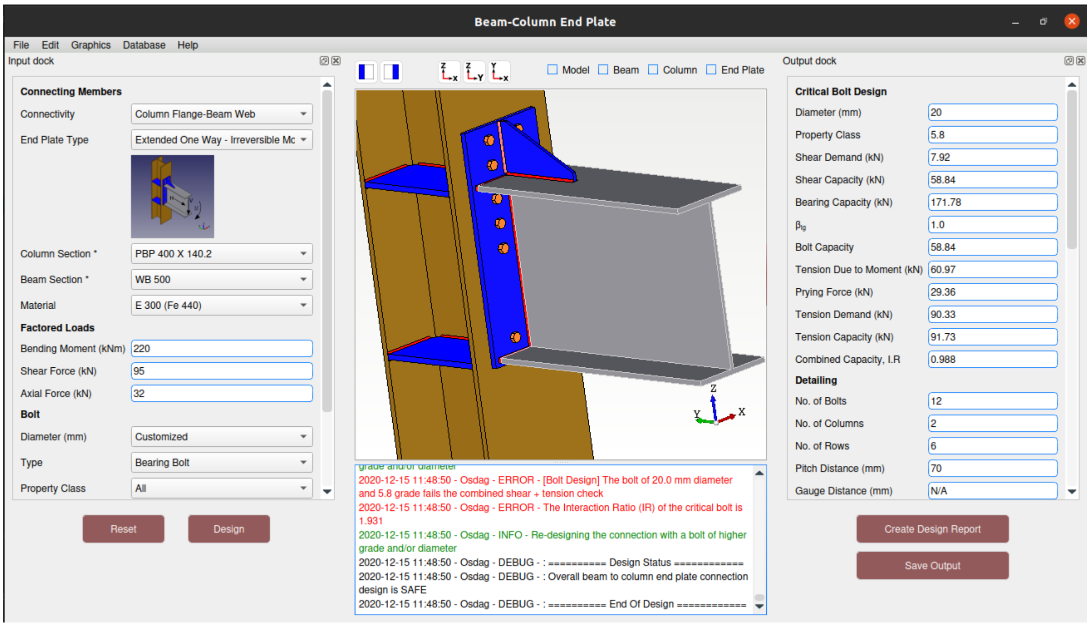
1.3 Osdag Software

Osdag (Open steel design and graphics) is a cross-platform, free/libre and open-source software designed for the detailing and design of steel structures based on the Indian Standard IS 800:2007. It allows users to design steel connections, members, and systems through an interactive graphical user interface (GUI) and provides 3D visualizations of designed components. The software enables easy export of CAD models to drafting tools for construction/fabrication drawings, with optimized designs following industry best practices [1, 2, 3]. Built on Python and several Python-based FLOSS tools (e.g., PyQt and PythonOCC), Osdag is licensed under the GNU Lesser General Public License (LGPL) Version 3.

1.3.1 Osdag GUI

The Osdag GUI is designed to be user-friendly and interactive. It consists of

- **Input Dock**: Collects and validates user inputs.
- **Output Dock**: Displays design results after validation.
- **CAD Window**: Displays the 3D CAD model, where users can pan, zoom, and rotate the design.
- **Message Log**: Shows errors, warnings, and suggestions based on design checks.



Connecting Members

Connectivity	Column Flange-Beam Web
End Plate Type	Extended One Way - Irreversible Mc
Column Section *	PBP 400 X 140.2
Beam Section *	WB 500
Material	E 300 (Fe 440)

Factored Loads

Bending Moment (kNm)	220
Shear Force (kN)	95
Axial Force (kN)	32

Bolt

Diameter (mm)	Customized
Type	Bearing Bolt
Property Class	All

Critical Bolt Design

Diameter (mm)	20
Property Class	5.8
Shear Demand (kN)	7.92
Shear Capacity (kN)	58.84
Bearing Capacity (kN)	171.78
β_{lg}	1.0
Bolt Capacity	58.84
Tension Due to Moment (kN)	60.97
Prying Force (kN)	29.36
Tension Demand (kN)	90.33
Tension Capacity (kN)	91.73
Combined Capacity, I.R	0.988

Detailing

No. of Bolts	12
No. of Columns	2
No. of Rows	6
Pitch Distance (mm)	70
Gauge Distance (mm)	N/A

```
grade and/or diameter
2020-12-15 11:48:50 - Osdag - ERROR - [Bolt Design] The bolt of 20.0 mm diameter
and 5.8 grade fails the combined shear + tension check
2020-12-15 11:48:50 - Osdag - ERROR - The Interaction Ratio (IR) of the critical bolt is
1.931
2020-12-15 11:48:50 - Osdag - INFO - Re-designing the connection with a bolt of higher
grade and/or diameter
2020-12-15 11:48:50 - Osdag - DEBUG - : ========== Design Status ============
2020-12-15 11:48:50 - Osdag - DEBUG - : Overall beam to column end plate connection
design is SAFE
2020-12-15 11:48:50 - Osdag - DEBUG - : ========== End Of Design ============
```

Figure 1.2: Osdag GUI

1.3.2 Features

- **CAD Model**: The 3D CAD model is color-coded and can be saved in multiple formats such as IGS, STL, and STEP.
- **Design Preferences**: Customizes the design process, with advanced users able to set preferences for bolts, welds, and detailing.
- **Design Report**: Creates a detailed report in PDF format, summarizing all checks, calculations, and design details, including any discrepancies.

For more details, visit the official Osdag website.

Chapter 2

Screening Task

2.1 Problem Statement

Development of windows installer

2.2 Tasks Done

In the winter internship I developed a windows installer for osdag. This installer was an online installer, means it required internet to install osdag on the system.

Chapter 3

Internship Task 1

3.1 Task 1: Help new interns

Continuing from my winter internship, during the initial weeks of the semester-long internship, I assisted new interns in setting up Osdag and resolving the issues they encountered. I also helped them become familiar with Osdag's philosophy and source code.

Chapter 4

Internship Task 2

4.1 Task 1: Resolve the Problems with the windows installer.

During testing it was found that the installer did not work on few computer on IIT Bombay computer lab's systems. The installer gets stuck or the system crashes. I was tasked to figure out why it was happening and how can we resolve it.

4.1.1 Found the problem

The installer required to have admin permission to run. Computers on IIT Bombay labs had non-admin users that's why the installations got stuck. The installer was not able to find miniconda installation because it was installed in system directories not in user directories.

4.1.2 Resolve

It was not necessary to ask for admin permissions to install osdag on user's machine so the permission level was changed from admin to user level. Directed users to install osdag, miniconda and miktex in user directories

Chapter 5

Internship Task 3

5.1 Task 1: Work with LCC Team

Osdag was to include two new optional plugins. I was tasked to work with the team and discuss on plugin building process. Few weeks we discussed on how plugins might function, how we can include it with the osdag. my primary task was to think about how these plugins can be installed along with osdag on users choice if they need them or not.

Chapter 6

Internship Task 4

6.1 Task 1: Find flaws in Conda based Installation

Osdag's development was initially carried out within a Conda environment due to the dependency on OpenCascade, which is originally developed in C++. Conda provides a convenient way to manage OpenCascade and its Python bindings, making it easier to integrate C++ libraries into the Python development workflow. As a result, Miniconda/Anaconda became a major dependency for Osdag, even for users who only wanted to work on steel structures and had no prior experience with Conda. Although efforts were made to simplify the Conda installation process for general users, the requirement of having Conda pre-installed on the system still posed significant usability challenges.

6.1.1 Findings

Osdag was originally designed as a Conda package, and the Windows installer simply automated the Conda installation process for the user. It retrieved the Conda executable path ($PROFILE/miniconda/Scripts/conda.exe$) to create a new Conda environment and install Osdag along with its dependencies in that environment. However, requiring users to have Miniconda and MiKTeX installed was a drawback, as they might never use these applications directly—they were only present due to Osdag's dependency on them.

6.2 Task 2: Find alternative to Conda based Installation

6.2.1 Solution

- Use python virtualenv
- Use Pixi enviornment

Python Virtualenv

Python's virtualenv creates isolated environments for projects, each with its own dependencies. This prevents conflicts between packages required by different projects. It ensures a clean setup and consistency across development and deployment. Environments can be created using $python - mvenvvenv\_name$ and activated as needed.

Pixi Environment

Pixi is a modern, fast package and environment manager designed for reproducible builds and lightweight virtual environments. Unlike traditional tools like Conda, Pixi uses a pixi.toml file to declaratively define environments, ensuring consistency across different systems. It supports creating isolated environments with specific Python versions and dependencies, and it is optimized for speed and reliability. Pixi is especially useful in projects where environment reproducibility and performance are critical.

How these methods will help?

One major drawback of using a Conda environment is that it is not relocatable—you can't simply copy the environment folder to another location or machine and expect it to work seamlessly. This is especially problematic when the environment contains shared libraries and Python bindings that rely on absolute paths. These dependencies often break if the environment is moved, leading to runtime errors or broken imports. In contrast, Python's venv environments are relatively more portable and simpler to manage, making them a better choice for distributing applications like Osdag to end users.

Chapter 7

Internship Task 5

7.1 Task 1: Migrate osdag from conda environment to python venv

To migrate from Conda to Python's venv, the Conda environment had to be replicated within a venv, ensuring it became relocatable. All of Osdag's dependencies had installed in this virtual environment so it could be bundled and distributed through a standalone installer (.exe) file. This approach allows users to run Osdag without needing Conda and distribute it easily.

Osdag relies on OpenCascade, a powerful C++ library used for 3D modeling, which is accessed in Python via its bindings. While Conda makes it convenient to install OpenCascade and its dependencies, it introduces challenges when it comes to portability. The Conda-based installation generates shared libraries and binaries with hardcoded, path-dependent links. As a result, relocating the environment to another system or directory often leads to broken paths and non-functional bindings. This lack of relocatability was a key motivation for moving to a Python venv setup, where more control can be exercised over the environment structure and deployment, especially for packaging with an installer.

7.2 Task 2: Build relocatable PythonOCC module

To compile PythonOCC so that it is relocatable and easily distributable, I first need to build OpenCASCADE with proper configuration to ensure its shared libraries have relative runtime paths. When configuring OpenCASCADE with CMake, set the CMAKE_INSTALL_PATH to a relative path such as *USERPROFILE* on Windows. This ensures that the OpenCASCADE shared libraries can be found relative to their installed location rather than absolute paths. Next, when building the pythonocc-core bindings, also configure CMake to use the OpenCASCADE installation directory and specify a relative PATH for the PythonOCC shared libraries, typically something like *USERPROFILE*/../lib, so that at runtime they can locate the OpenCASCADE libraries relative to their own location. After installation, package the entire directory structure, including both the OpenCASCADE shared libraries and the PythonOCC Python modules, preserving their relative layout. Because of the relative PATH settings, one can move or distribute this entire folder anywhere on a target system without needing to reset environment variables or reinstall dependencies. Users simply need to add the PythonOCC module folder to their PYTHONPATH, and the shared libraries will load correctly thanks to the relative PATH, enabling seamless usage across different machines or directory locations. This method ensures a fully relocatable and portable PythonOCC build that simplifies deployment and distribution.

Chapter 8

Internship Task 6

8.1 Task 1: Use Pixi Setup for Osdag Installer

While I was working on compiling a relocatable PythonOCC module to include in a Python virtual environment for creating a portable Osdag setup, Athithya—another intern—was exploring an alternative approach using Pixi. He discovered a simpler and more efficient way to build a relocatable environment for Osdag. By using Pixi, we could easily recreate the environment, manage and update dependencies, and distribute it across systems. The setup required only a project.toml file, which serves as the configuration for the Pixi environment. Pixi creates a lightweight, relocatable environment similar to Conda, but without the need to install Miniconda on the machine. This made the process much more streamlined and user-friendly, offering an elegant solution for maintaining and sharing the Osdag environment. See Appendix B, Winodws Installer for Osdag (Developer's Guide).

8.2 Task 2: Include Steve's Osdag_latex_env

Another intern was working on removing the dependency on MiKTeX from Osdag. To achieve this, he proposed a solution to bundle the LaTeX executable directly with the Osdag installation. This approach eliminates the need for users to have a separate pdflatex installation on their machines. By packaging a minimal, portable LaTeX distribution along with Osdag, the application can generate PDFs internally without relying on external LaTeX tools. This significantly simplifies the setup process for end-users and ensures consistent PDF generation across different systems. See appendix C, Winodws Installer for Osdag (Developer's Guide).

New installer for the osdag has to incorporate both these features to remove conda and miktex dependence from osdag.

Chapter 9

Internship Task 7

9.1 Task 1: Recreate the Installer

New osdag installer was to be created such that it bundles entire pixi environment and extract it on the user's machine on their desired location.

9.2 Tasks Done

9.2.1 Create Pixi environment

To achieve this, I first had to create the latest Pixi environment for the most recent release of Osdag using a properly configured pixi.toml file. This file defined all the necessary dependencies and configurations required to run Osdag,, See appendix B, Windows Installer for Osdag (Developer's Guide).

9.2.2 Ensure the latex is included with the environment

This environment still required MiKTeX or another external LaTeX tool to generate PDFs. To fully eliminate this dependency, I had to include Steve's standalone LaTeX environment as part of the Osdag setup. By bundling Steve's minimal and portable LaTeX environment, Osdag can now compile LaTeX documents and generate PDFs internally, without relying on any LaTeX installation on the user's machine. This makes the environment completely self-contained and significantly improves portability and ease of use. See appendix C, Windows Installer for Osdag (Developer's Guide).

9.2.3 Create NSIS Script

With the environment now fully set up, I proceeded to write the NSIS (Nullsoft Scriptable Install System) script to create the Osdag installer. This script defines the behavior and appearance of the installer, including how it interacts with the user during installation. It handles downloading or extracting the bundled Osdag (along with the Pixi environment) into the user's chosen installation directory. Additionally, the script creates convenient desktop and Start Menu shortcuts, enabling users to launch Osdag with a single click. This ensures a smooth, user-friendly installation experience and makes Osdag easily accessible post-installation.

The code and explanation of the code is already given in the Developer's guide for recreating this installer in future, i have created.

Chapter 10

Internship Task 8

10.1 Task 1: Fix some problems with the new installer

- Uninstaller doesn't check if osdag is closed, Uninstaller doesn't confirm. It should also show that ”everything here will be deleted”.
- There is a terminal when running osdag. It also doesn't automatically close with Osdag.
- No ”hide details” when running installer, There is no option to ”launch osdag” upon installation.
- Confirmation of final folder path should be right on the same page (no ”go back to change”).

10.1.1 Tasks Done

- Added a confirmation pop-up if the user hits uninstall osdag.
- The terminal was made hidden using nircmd (See Developer's Guide).
- Added a Launch Osdag checkbox at th finish page os installer.
- I couldn't find any way of hide/show details toggle.

10.2 Task 2: Detected as Virus

The installer was found to detected as virus by the system's anti-virus software, this is because the installer was not digitally signed.

10.2.1 How to fix ?

We need to get a digital certificate for the installer and sign each version of the installer digitally using windows signtool. See Developer's Guide.

10.3 Task 3: Prepare Final Insatller for release, document the Developer's Guide

As my internship drew to a close, I focused on finalizing the offline installer by addressing a few remaining minor issues to ensure it was ready for testing before the official release. In parallel, I began documenting the entire process of creating the installer—from setting up the Pixi environment and bundling Steve's LaTeX setup to writing the NSIS script and configuring the installer behavior. This documentation was compiled into a comprehensive Developer's Guide aimed at facilitating the creation of future installers. It serves as a step-by-step reference for other developers, helping ensure continuity and ease of maintenance in Osdag's deployment process.

10.4 Task 4: Include Plugin installations

Gave an idea of how plugins can be installed along with osdag See appendix D, Windows Installer for Osdag (Developer's Guide).

Chapter 11

Internship Task 8

11.1 Task 1: A maintainer's manual for Osdag Installer

This mannual is included in the developer's guide created.

11.2 Task 2: A Developer's Guide for Osdag Installer

This guide provides a comprehensive overview of the current installer's workflow, including detailed code explanations and the rationale behind key development decisions made throughout the process.

11.3 Task 3: A framework on installation of plugins through the installer

See Appendix D, Windows installer for Osdag(Developer's Guide). The framework with proof of work is already included in the document.

11.4 Task 4: A framework to incorporate unit tests in the installer

Chapter A

Appendix

A.1 Work Reports

DATE	DAY	TASK	# Hours worked
20 Feb 2025	Thursday	Helped Interns to setup osdag	-
21 Feb 2025	Friday	Helped Interns to setup osdag	-
24 Feb 2025	Monday	Helped Interns to setup osdag	-
25 Feb 2025	Tuesday	Helped Interns to setup osdag	-
26 Feb 2025	Wednesday	Helped Interns to setup osdag	
27 Feb 2025	Thursday	Helped Interns to setup osdag	1
28 Feb 2025	Friday	Helped Interns to setup osdag	2
3 Mar 2025	Monday	Worked with LCC team and Installer team	2
4 Mar 2025	Tuesday	Worked with LCC team and Installer team	3
5 Mar 2025	Wednesday	Worked with LCC team and Installer team	3
6 Mar 2025	Thursday	Worked with LCC team and Installer team	2
7 Mar 2025	Friday	Worked with LCC team and Installer team	3
10 Mar 2025	Monday	Discussed Commonn problems with the conda based installer,did initial research to fix t	3
11 Mar 2025	Tuesday	Research on conda based installer problems	2
12 Mar 2025	Wednesday	Research on conda based installer problems	2
13 Mar 2025	Thursday	Research on conda based installer problems	1
14 Mar 2025	Friday	Prepared document for step by step installer guide for other interns	3
17 Mar 2025	Monday	Discussed upon pixi and inno setup fro creating installer	2
18 Mar 2025	Tuesday	Read on inno setup, conda based installer being detected as virus	2
19 Mar 2025	Wednesday	read on pixi , prepared document on registry keys	3
20 Mar 2025	Thursday	Made suggested changes in the conda based installer	4
21 Mar 2025	Friday	Made suggested changes in the conda based installer No admin privileges needed., installer header image resized., improved dialogue boxes., added custom directory selection page added for miniconda, More guided messages for user, user can choose the existing miniconda folder or can give a folder in which miniconda w	3
26 Mar 2025	Wednesday	How conda env can be made localizable	3
27 Mar 2025	Thursday	How conda env can be made localizable	4
28 Mar 2025	Friday	Tried compile OCC module from scratch	4
31 Mar 2025	Monday	Tried compile OCC module from scratch	4
1 Apr 2025	Tuesday	Relocalizable binaries for OCC	4
2 Apr 2025	Wednesday	looked into Steve's localized latex env	2
3 Apr 2025	Thursday	Looked into Athithya's pixi env setup for osdag	3
4 Apr 2025	Friday	Incorporated Athithya's and steve's work in installer	3
7 Apr 2025	Monday	Updated athithya's run and update batch scripts	3
8 Apr 2025	Tuesday	Created new shortcuts for launching osdag	3
9 Apr 2025	Wednesday	Completed the first offline installer for osdag	4
10 Apr 2025	Thursday	Testing of new pixi based installer	2
11 Apr 2025	Friday	Testing of new pixi based installer	2

DATE	DAY	TASK	Hours worked
14 Apr 2025	Monday	Uninstaller doesn't check if osdag is closed, Uninstaller doesn't confirm. It should also show that "everything here will be deleted", There is a terminal when running osdag. It also doesn't automatically close with Osdag., No "hide details" when running installer, There is no option to "launch osdag" upon installation, Confirmation of final folder path should be right on the same page (no "go back to chang	3
15 Apr 2025	Tuesday	Worked on fixing the above issues	4
16 Apr 2025	Wednesday	Worked on fixing the above issues	4
17 Apr 2025	Thursday	Read on digital signing of installer	3
18 Apr 2025	Friday	updated pixi based installer with steve's updated work	4
21 Apr 2025	Monday	How to sign osdag installer	2
22 Apr 2025	Tuesday	Leave	-
23 Apr 2025	Wednesday	Leave	-
2 May 2025	Friday	Fix installer header image	2
5 May 2025	Monday	Followed on Athithya's work on plugins	1
6 May 2025	Tuesday	Followed on Athithya's work on plugins	2
7 May 2025	Wednesday	Followed on Athithya's work on plugins	1
8 May 2025	Thursday	Hide the terminal of osdag	2
9 May 2025	Friday	Refine the installer fix issues logic and comments	4
12 May 2025	Monday	Finalize the Installer	4
13 May 2025	Tuesday	Uploaded Final pixi based offline instaler for testing	1
14 May 2025	Wednesday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
15 May 2025	Thursday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
16 May 2025	Friday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
17 May 2025	Saturday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
18 May 2025	Sunday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
19 May 2025	Monday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
20 May 2025	Tuesday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
21 May 2025	Wednesday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
22 May 2025	Thursday	LEAVE (working on installer developers guide, Attende meets for making plugin)	4
23 May 2025	Friday	LEAVE (working on installer developers guide, Attende meets for making plugin)	4
24 May 2025	Saturday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
25 May 2025	Sunday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
26 May 2025	Monday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
27 May 2025	Tuesday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
28 May 2025	Wednesday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
29 May 2025	Thursday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
30 May 2025	Friday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
31 May 2025	Saturday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
1 Jun 2025	Sunday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-

DATE	DAY	TASK	# Hours worked
2 Jun 2025	Monday	LEAVE (working on installer developers guide, Attende meets for making plugin)	-
3 Jun 2025	Tuesday	Prepare Maintainers Mannual for installer	4
4 Jun 2025	Wednesday	Work on Test Suite for installer	3
5 Jun 2025	Thursday	Work on Test Suite for installer	3
9 Jun 2025	Monday	Fix installer App icon not showing	2
10 Jun 2025	Tuesday	Resolve pdf generation issue on system's with multiple users	2
11 Jun 2025	Wednesday	Work on Test Suite for installer	2
12 Jun 2025	Thursday	Work on Test Suite for installer	2

Winodws Installer for Osdag (Developer's Guide)

Submitted by

Mehendi Hasan
3rd Year BSc Student, Physics Major, Computer Science Minor
University of Delhi

Under the Guidance of

Prof. Siddhartha Ghosh
Department of Civil Engineering
Indian Institute of Technology Bombay

Mentors:

Ajmal Babu M S
Parth Karia
Ajinkya Dahale

June 14, 2025

Contents

Chapter 1

Introduction

1.1 Conda and Pixi based Installations

Osdag's development was initially carried out within a Conda environment due to the dependency on OpenCascade, which is originally developed in C++. Conda provides a convenient way to manage OpenCascade and its Python bindings, making it easier to integrate C++ libraries into the Python development workflow. As a result, Miniconda/Anaconda became a major dependency for Osdag, even for users who only wanted to work on steel structures and had no prior experience with Conda. Although efforts were made to simplify the Conda installation process for general users, the requirement of having Conda pre-installed on the system still posed significant usability challenges.

As an alternative to the Conda-based environment, the Pixi environment has been chosen for use. Pixi is a modern package and environment manager that offers several key advantages, including faster environment creation, support for reproducible builds, and native integration with pyproject.toml. One of its most significant features is the ability to create lightweight and relocatable environments that can be packaged and distributed without requiring global installation. This makes Pixi an ideal choice for bundling Osdag with all its dependencies, enabling easier installation and use for end-users without prior experience in environment management.

Pixi environments are typically stored within the .pixi/envs directory of a project, keeping them isolated and easy to manage. This design facilitates portability and cleanup. While Conda environments are generally not relocatable, Pixi addresses this limitation. Pixi emphasizes a project-centric approach, storing environments locally within the project directory. This strategy avoids polluting the global environment and ensures that each project remains self-contained. Such isolation not only keeps the system clean but also simplifies dependency management and reduces overhead.

1.2 Current Conda based Installer

The current Windows installer for Osdag was designed around a Conda-based installation workflow, which required Conda to be pre-installed on the user's machine. During the installation process, the installer checks for an existing Conda setup and, if not found, proceeds to install Miniconda automatically. It then creates a new Conda environment and installs Osdag along with its dependencies within that environment. Installing Miniconda on the user's machine was an extra step we aimed to eliminate. Moreover, the Miniconda installation frequently failed on many systems due to insufficient permissions or outdated software environments. During testing, the team discovered that the installer did not function correctly on the lab computers at IIT Bombay. As a result, addressing this issue became a priority.

1.3 Solution for the above problems

Moving from a Conda-based environment to a Pixi-based environment helped us overcome several challenges we previously faced:

- Eliminated the dependency on Conda
- Enabled creation of relocatable environments
- Simplified the NSIS installer script
- Made it possible to build a fully offline installer
- Incorporated plugin installation along with Osdag installation

The new installer also removes the need for users to have MiKTeX or pdflatex preinstalled on their system to generate reports. Instead, pdflatex is bundled within the osdag environment itself, ensuring seamless report generation without additional configuration. Additionally, the installer creates desktop and Start Menu shortcuts, making it easier for users to launch Osdag and enhancing the overall user experience. It also Creates a unistaller file for the Osdag application.

Chapter 2

New Installer

2.1 How this installer works

With pixi environment, it was easier to relocate the entire environment to a new location and the application will work fine; this new installer takes advantage of it. The previous installer would create a new conda environment and install osdag along with its dependencies in that environment that required Internet connection on users machine. But the new installer is bundled with the pixi environment, and it extracts the environment to the desired location-Osdag installation directory.

2.2 NSIS Script for the installer

2.2.1 Folder Structure

```
|  Pixi-installer
|   |__version25.01.0.2-win
|       |__Osdag-2025.01.0.2-win.nsi
|       |__Osdag-v2025.01.0.2-win.exe
|__osdag-pixi
|   |__.pixi
|   |   |__envs
|   |       |__default
|   |__icons
|   |   |__Osdag.ico
|   |   |__Osdag_App_icon.ico
|   |__pixi.exe
|   |__pixi.lock
|   |__pixi.toml
|__license.txt
|__Osdag.ico
|__Osdag_App_icon.ico
```

```
├── Osdag_header.bmp
├── run.bat
└── update.bat
```

version25.01.0.2-win: contains the nsis script, and the installer executable is created here.
osdag-pixi: This folder contains the pixi environment that was created with the pixi.toml file. pixi,exe is the executable required to run pixi command, the pixi.toml file contains the configuration for creating the pixi environment. It also generates a pixi.lock file of the dependencies and their specific versions. See Appendix B for how to create pixi environments.

license.txt: lgpl 3 license for the osdag application.
run.bat: batch script to run osdag.(See Appendix A)
update.bat: batch script to update osdag version. (See Appendix A)
nircmd.exe: It lets us perform a wide variety of Windows tasks silently and efficiently — without opening a console window or GUI prompt. ’nircmd.exe exec hide ”*INSTDIR/Osdag/run.bat*”’ Run run.bat in the background (hidden), with no command prompt popping up.

2.3 NSIS Script Code and Explanation

Download the File from github repository Zehen-249/Osdag-installer

```
!define APP_NAME "Osdag"
!define APP_VERSION "2025.01.0.2"
!define APP_PUBLISHER "Osdag Team IIT Bombay"

OutFile "Osdag-v${APP_VERSION}-win.exe"
Name "${APP_NAME}"
VIProductVersion "${APP_VERSION}"
VIAddVersionKey "ProductName" "${APP_NAME}"
VIAddVersionKey "FileVersion" "${APP_VERSION}"
VIAddVersionKey "FileDescription" "Osdag version ${APP_VERSION} windows Installer"
VIAddVersionKey ’LegalCopyright’ ’c 2025 Osdag Team, IIT Bombay’

RequestExecutionLevel user

!include "MUI2.nsh"
!include "nsDialogs.nsh"
!include "LogicLib.nsh"

!define MUI_WELCOMEPAGE_TITLE "This Setup will guide you through the installation of Osdag $\r$\n$\r$\nPLEASE UNINSTALL ANY EARLIER VERSION OF OSDAG on your system before going ahead.$\r$\n $\r$\nPlease click Next only after uninstalling the earlier version"
!define MUI_FINISHPAGE_TITLE "Installation Complete"
!define MUI_FINISHPAGE_TEXT "Osdag has been successfully installed."
!define MUI_ABORTWARNING
```

```
!define MUI_ICON "..\Osdag.ico"
!define MUI_UNICON "..\Osdag.ico"
!define MUI_HEADERIMAGE
!define MUI_HEADERIMAGE_BITMAP "..\Osdag_header.bmp"
!define MUI_HEADERIMAGE_BITMAP_NOSTRETCH
!define MUI_HEADERIMAGE_UNBITMAP "..\Osdag_header.bmp"
!define MUI_HEADERIMAGE_UNBITMAP_NOSTRETCH
!define MUI_LICENSEPAGE_CHECKBOX
!define MUI_LICENSEPAGE_CHECKBOX_TEXT "I accept the terms in the license agreement"
!define MUI_DIRECTORYPAGE_TEXT_TOP "Select Installation Directory"
!define MUI_COMPONENTSPAGE_TEXT_TOP "Check the plugins you want to Install and uncheck those you don’t want to install. Click Next to continue."
!define MUI_COMPONENTSPAGE_TEXT_COMPLIST "Select additional plugins to install:"
!define MUI_FINISHPAGE_RUN
!define MUI_FINISHPAGE_RUN_TEXT "Launch Osdag"
!define MUI_FINISHPAGE_RUN_FUNCTION LaunchOsdag
!define MUI_UNFINISHPAGE_TITLE "Uninstallation Complete"
!define MUI_UNFINISHPAGE_TEXT "Osdag was successfully removed from your system."

!insertmacro MUI_PAGE_WELCOME
!insertmacro MUI_PAGE_LICENSE "..\license.txt"
!insertmacro MUI_PAGE_COMPONENTS
!insertmacro MUI_PAGE_DIRECTORY
page custom DirectoryConfirmationPage
!insertmacro MUI_PAGE_INSTFILES
!insertmacro MUI_PAGE_FINISH
!insertmacro MUI_UNPAGE_INSTFILES
!insertmacro MUI_UNPAGE_FINISH
!insertmacro MUI_LANGUAGE "English"

Function .onInit
    StrCpy $INSTDIR "$PROFILE\Osdag"
FunctionEnd

Function DirectoryConfirmationPage

    nsDialogs::Create 1018
    Pop $0

    ${If} $0 == error
        Abort
    ${EndIf}
    ${NSD_CreateLabel} 0 0 100% 40u "Osdag will be installed at:$\r$\n$INSTDIR\Osdag$\r$\n$\r$\nIf you want to change the location, click Back to return to the directory selection page."
    Pop $1
    nsDialogs::Show

FunctionEnd

Section "Osdag" SEC_Main
    SectionIn RO
    SetOutPath "$INSTDIR\Osdag"

    ; copy files to install\Osdag folder
    File /r "..\osdag-pixi\.pixi"
    File /r "..\osdag-pixi\icons"
    File "..\license.txt"
    File "..\run.bat"
    File "..\update.bat"
    File "..\nircmd.exe"

    ; create uninstaller for osdag
    WriteUninstaller "$INSTDIR\Osdag\Uninstall.exe"

    ; Registry keys
    WriteRegStr HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "DisplayName" "${APP_NAME}"
    WriteRegStr HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "Publisher" "${APP_PUBLISHER}"
    WriteRegStr HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "DisplayVersion" "${APP_VERSION}"
    WriteRegStr HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "UninstallString" "$INSTDIR\Osdag\Uninstall.exe"
    WriteRegStr HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "InstallLocation" "$INSTDIR\Osdag"
    WriteRegDWORD HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "NoModify" 1
    WriteRegDWORD HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "NoRepair" 1

SectionEnd

; optional
Section "Startmenu Shortcut" SEC_SM_Shortcuts
    CreateDirectory "$SMPROGRAMS\Osdag"

    CreateShortcut "$SMPROGRAMS\Osdag\Osdag.lnk" "$INSTDIR\Osdag\nircmd.exe" ’exec hide "$INSTDIR\Osdag\run.bat"’ "$INSTDIR\Osdag\icons\Osdag_App_icon.ico" 0

    CreateShortcut "$SMPROGRAMS\Osdag\Uninstall.lnk" "$INSTDIR\Osdag\Uninstall.exe" "" "$INSTDIR\Osdag\icons\Osdag_App_icon.ico" 0
SectionEnd

; optional
Section "Desktop Shortcut" SEC_DESK_Shortcuts

    CreateShortcut "$DESKTOP\Osdag.lnk" "$INSTDIR\Osdag\nircmd.exe" ’exec hide "$INSTDIR\Osdag\run.bat"’ "$INSTDIR\Osdag\icons\Osdag_App_icon.ico" 0

SectionEnd


; optional
Function LaunchOsdag
    Exec ’"$INSTDIR\Osdag\nircmd.exe" exec hide "$INSTDIR\Osdag\.pixi\envs\default\
            Scripts\osdag.exe"'
116 FunctionEnd
117
118 Function un.onInit
119     MessageBox MB_YESNO|MB_ICONQUESTION "Are you sure you want to uninstall Osdag?"
            IDYES +2
120     Abort
121     ReadRegStr $INSTDIR HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\Osdag"
             "InstallLocation"
122 FunctionEnd
123
124 Section "Uninstall"
125
126     RMDir /r $INSTDIR
127
128     Delete "$DESKTOP\Osdag.lnk"
129     Delete "$SMPROGRAMS\Osdag\Osdag.lnk"
130     Delete "$SMPROGRAMS\Osdag\Uninstall.lnk"
131     RMDir /r "$SMPROGRAMS\Osdag"
132
133     DeleteRegKey HKCU "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}"
134
135 SectionEnd
```

Listing 2.1: NSIS Installer Script

2.4 Explaining the Code

2.4.1 Line 1-3

set variables

2.4.2 Line 5-11

- set installer file name
- set application name
- set product version, name, company name, and description. version should follow X.X.X.X format.

2.4.3 Line 13

set execution permission to user-level.

2.4.4 Line 15-17

Include required libraries/packages. Make these packages are downloaded with your NSIS.

2.4.5 Line 19-39

These lines configure the user interface and messages for the Osdag installer and uninstaller using NSIS. They set the welcome message urging users to uninstall previous versions before proceeding, define titles and texts for the installation and uninstallation completion pages, and include icons and header images for the installer windows. The script enables a license agreement checkbox, customizes text for directory selection and plugin component pages, and adds an option to launch Osdag immediately after installation. Additionally, it ensures the user receives a warning if they abort the installation process.

2.4.6 Line 41-50

- **!insertmacro MUI_PAGE_WELCOME**: Adds a Welcome page at the start of the installer, greeting the user.
- **!insertmacro MUI_PAGE_LICENSE "../license.txt"** : Adds a License Agreement page showing the license text loaded from the specified file path.
- **!insertmacro MUI_PAGE_COMPONENTS**: Adds a Component Selection page where users can choose which parts/features of the software to install.
- **!insertmacro MUI_PAGE_DIRECTORY**: Adds a Directory Selection page where users specify the installation folder.
- **page custom DirectoryConfirmationPage**: Inserts a custom page named DirectoryConfirmationPage at this point in the installation sequence (for confirming the directory or extra info).
- **!insertmacro MUI_PAGE_INSTFILES**: Adds the Installation Progress page that shows the status of the files that are being installed.
- **!insertmacro MUI_PAGE_FINISH**: Adds the Finish page at the end of the installation to indicate completion.
- **!insertmacro MUI_UNPAGE_INSTFILES**: Adds the Uninstaller Progress page that shows progress when uninstalling the program.
- **!insertmacro MUI_UNPAGE_FINISH**: Adds the Uninstaller Finish page to indicate that the uninstall process is complete.
- **!insertmacro MUI_LANGUAGE "English"**: Sets the installer language to English.

2.4.7 Line 52-54

This defines a special callback function that runs at the very start of the installer, before any pages are shown. This sets the default installation directory ($INSTDIR$) to a folder named Osdag inside the user's profile directory ($PROFILE$).

2.4.8 Line 56-68

This NSIS function *DirectoryConfirmationPage* defines a custom installer page that displays a confirmation message about the selected installation directory. Users see a message like:
Osdag will be installed at: C:/Users/*PROFILE*/Osdag If you want to change the location, click Back to return to the directory selection page.

2.4.9 Line 70-94

This section installs the Osdag app, sets up required files, creates an uninstaller, and registers it properly in the system's ”Add/Remove Programs” panel. Section ”Osdag” SEC_Main defines the installation section. SectionIn RO makes it required (user can't deselect it). SetOutPath ”*INSTDIR/Osdag*” sets the install location to a subfolder named Osdag in the chosen directory. Copies these files into the install folder: - The .pixi environment (recursively) - license.txt, run.bat, update.bat, nircmd.exe. Generates an uninstaller named uninstaller.exe inside the install folder.
Windows Registry entries (for Programs and Features): Creates registry keys under *HKCU/.../Uninstall*/APP_NAME so the app appears in ”Add or Remove Programs”: - DisplayName: Shows name as ”Osdag” - UninstallString: Points to uninstaller executable - DisplayIcon: Uses Osdag icon from the Pixi environment - InstallLocation: Sets the install directory

2.4.10 Line 97-116

Adds optional functionality to create shortcuts and launch Osdag silently. The ”Desktop Shortcut” section creates a desktop shortcut that launches Osdag silently via nircmd.exe and run.bat. The ”Startmenu Shortcut” section creates a Start Menu folder named ”Osdag” and places two shortcuts inside it: one to launch Osdag silently using nircmd.exe with run.bat, and another for the uninstaller, both using a custom icon from the Pixi environment. Additionally, the LaunchOsdag function defines a command to directly run the Osdag executable inside the Pixi environment in hidden mode, allowing for silent execution during or after installation. This setup improves usability by giving users quick access to launch and uninstall Osdag, while keeping the interface clean and background processes hidden.

2.4.11 Line 118-136

Defines the uninstallation behavior for Osdag. The *un.onInit* function first prompts the user with a confirmation message asking if they really want to uninstall Osdag. If the user selects ”No”, the process is aborted; if ”Yes”, it proceeds by reading the install location from the Windows Registry. The ”Uninstall” section then performs the cleanup: it recursively deletes the Osdag installation directory, removes the desktop and Start Menu shortcuts, and deletes the Start Menu folder itself. Finally, it removes the registry entry that made Osdag appear in the ”Add or Remove Programs” list. This ensures a complete and clean uninstallation of the application.

2.4.12 Additionally

Developer needs to Add some additional files and folders manually to their places inside osdag source code under site-packages until the conda channel source code is not updated. These files and folder are:

- Place osdag-latex-env folder into ResourceFiles (See Appendix C)
- Replace SectionModular_latex.py file inside site-packages/Osdag (See Appendix C)
- Replace reportGenerator_latex.py file inside site-packages/Osdag/design_report (See Appendix C)

Plugins

The new installer uses the components approach where each section of the installer is a required or optional component. User has choice to select the components while installing osdag and only those components will be installed. In the future versions of the installer, optional sections for each new plugin can be included. (See Appendix D)

Chapter 3

Recreate Installer

Following are the steps to recreate the folder structure to start working on the installer,

- Fork this repository Zehen-249/Osdag-installer, main branch. Go thorugh the README.md.
- Create the Pixi environment. (See Appendix B)
- Paste the files and folders if not already present in the environment as directed above. (See Chapters 2 Additionally)
- Install NSIS (See Appendix E)
- Create a new folder versionX.X.X.X-win for new version of installer.
- Make the necessary changes in the Installer Script for the new version. Change path for bundling the new Pixi environment. (See README.md in the repository)
- Compile the Script. (See Appendix E)

Chapter 4

Maintainers Guide

4.1 Setup

4.1.1 Clone the installer branch

```
git clone --branch main --single-branch https://github.com/Zehen-249/Osdag-Installer.git
```

4.1.2 Recreate the pixi environment

- Open terminal and go to osdag-pixi folder ‘cd */path/to/pixi-installer/osdag-pixi‘

4.1.3 To update the installer script

- Run ‘pixi install‘
- It will Read the pixi.lock file.
- Create a deterministic environment with exact versions and platform-specific builds

4.1.4 To update Osdag with the latest release version on conda channel or any dependency of Osdag

- Edit the ‘pixi.toml‘ file.
- Specify the updated version of the dependency or Osdag.
- Now run ‘pixi install‘.
- It will regenerate pixi.lock file with update versions and will create the environment.

4.2 Maintenance

- Once the pixi environment for the latest release is created, create a new folder in the parent directory for the new release version of the installer, follow the pattern.
- Update the outputfile name as as needed, follow the pattern.
- Update product version.
- Make the neccessary changes in the installer script (.nsi).
- Compile the script. (See Appendix E)
- The installer.exe will be created in the same directory where .nsi script is placed.
- Get this new installer digitally signed with the certificate. (See Chapter 5)
- Distribute this new installer.

Update this document if necessary for future maintainers.

Chapter 5

Digital Signing of Installer

5.1 What is Digital Signature

A digital signature is a cryptographic method used to verify the authenticity and integrity of digital data — like emails, documents, or software. It’s the digital equivalent of a handwritten signature or a stamped seal, but much more secure.
To avoid the ”Unauthorized Publisher” or ”Unknown Publisher” warning when running the installer on Windows, we need to digitally sign the installer executable using a code signing certificate.

5.1.1 Each Version Must Be Signed

Every time we rebuild the installer (a new version of Osdag), the resulting .exe file has a different binary hash, even if most of the content is the same. Since the digital signature is tied to the exact binary, a new signature is required for each version. Use the same code signing certificate to sign each new installer build.
We do not need to buy a new certificate for each version. A single valid certificate can be reused for all builds during its validity period.

5.1.2 How to get signed?

- Step 1: Purchase a Code Signing Certificate
  Choose a Certificate Authority (CA) - DigiCert, GoDaddy, GlobalSign.
  Receive certificate in .pfx format (or .spc and .pvk).

- Step 2: Set Up the Tools
  Install Windows SDK
  During installation, select “Signing Tools for Desktop Apps”.
  Locate signtool.exe, Usually found in $C : /ProgramFiles(x86)/WindowsKits/10/bin/ < version > /x64/signtool.exe$

- Step 3: Sign The Installer
  Run this command

```
    "C:\Program Files (x86)\Windows Kits\10\bin\<version>\x64\signtool.exe" sign ^
        /f "C:\path\to the\certificate\code-signing.pfx" ^
        /p your_pfx_password ^
        /tr http://timestamp.digicert.com ^
        /td sha256 ^
        /fd sha256 ^
        "C:\Path\To\Your\installer.exe"
```

Appendix A

Run.bat and Update.bat Batch Scripts(Athithya’s Work)

A.1 Run.bat

This is the batch script to launch osdag from the pixi environment. It reads the Osdag insatallation directory from registry key entry of osdag and uses this path to call Osdag.exe.

```
@echo off
setlocal enabledelayedexpansion

:: Query the registry for Osdag install location (check HKCU)
for /f "tokens=3*" %%A in (’reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Uninstall\Osdag" /v InstallLocation 2^>nul’) do (
    set "INSTALL_PATH=%%A %%B"
)

:: Trim trailing space if present
set "INSTALL_PATH=!INSTALL_PATH:~0,-1!"

:: Check if the InstallLocation was found
if defined INSTALL_PATH (
    :: Set the environment path (with quotes to handle spaces correctly)
    set "ENV_PATH=!INSTALL_PATH!\.pixi\envs\default"

    :: Check if osdag.exe exists
    if not exist "!ENV_PATH!\Scripts\osdag.exe" (
        pause
        exit /b
    )

    :: Run Osdag
    cmd /k ’set OSDAG_LAUNCHED_FROM=launcher && \"!ENV_PATH!\Scripts\osdag.exe"’
) else (
    exit /b
)
```

```

endlocal
pause
```

Listing A.1: Run.bat

A.2 update.bat

```
@echo off
setlocal enabledelayedexpansion

:: Query the registry for Osdag install location (check HKCU)
for /f "tokens=3*" %%A in ('reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Uninstall\Osdag" /v InstallLocation 2^>nul') do (
    set "INSTALL_PATH=%%A %%B"
)

:: Trim trailing space if present
set "INSTALL_PATH=!INSTALL_PATH:~0,-1!"

:: Check if the InstallLocation was found
if defined INSTALL_PATH (
    echo Osdag install path: !INSTALL_PATH!

    :: Change to the Osdag installation directory
    cd /d "!INSTALL_PATH!"

    :: Use local pixi.exe if available
    if exist "pixi.exe" (
        set "PIXI_CMD=pixi.exe"
    ) else (
        set "PIXI_CMD=pixi"
    )

    :: Check for updates
    echo Checking for Osdag updates...
    %PIXI_CMD% update

    if errorlevel 1 (
        echo.
        echo Update failed. Possible reasons:
        echo 1. No internet connection
        echo 2. Server unavailable
        echo 3. Installation corrupted
        echo.
        echo Try running: %PIXI_CMD% clean && %PIXI_CMD% install
        pause
        exit /b 1
    )

    :: Show updated version
    echo.
    echo Update successful! Current version:
    %PIXI_CMD% list | findstr osdag

) else (
    echo Osdag is not installed or registry key not found.
)

pause
exit /b 0
```

Listing A.2: update.bat

Appendix B

Pixi Installation

B.1 Introduction

Pixi is a modern environment manager that is fast, lightweight, and easy to use. Unlike Miniconda, which has a large installation size (over 400 MB) and can be slow with complex dependencies, Pixi is much smaller (around 15–20 MB) and works faster. It uses simple configuration files, supports quick setup, and helps maintain reliable and consistent environments.

B.2 Purpose of Pixi

Pixi is primarily used for:

- Configuration: Utilizes a single `pixi.toml` file to define all project dependencies, ensuring consistency across environments.
- Command Execution: Streamlines environment setup and task execution with simple commands like `pixi add`, `pixi run`, and `pixi shell`.
- Reproducible Environments: Generates a `pixi.lock` file to ensure environments can be reliably reproduced across systems.
- Dependency Management: Offers faster resolution and installation times compared to Miniconda.
- Multi-Environment Support: Allows defining multiple environments (e.g., `dev`, `test`) within the same project.
- Task Support: Enables task automation (e.g., build, test) via named tasks in `pixi.toml`.
- Cross-Platform Compatibility: Works uniformly across Linux, macOS, and Windows.

B.3 Understanding the `pixi.toml` File

The `pixi.toml` file is the central configuration for the project. It contains metadata, dependencies, tasks, and environment definitions.

B.3.1 Key Sections

- `[workspace]`: Defines project name, version, channels, authors, and platform targets.
- `[dependencies]`: Lists global dependencies.
- `[feature.<name>.dependencies]`: Lists dependencies specific to a feature (e.g., `editable`, `noneditable`).
- `[feature.<name>.tasks]`: Declare tasks that can be executed conditionally based on the active feature (environment).
- `[environments]`: Defines named environments and associates them with features.

B.3.2 Sample `pixi.toml` file

Below is the actual `pixi.toml` file used in the Osdag project:

```
[project]
name = "osdag"
version = "2025.01.a.2"
description = "Osdag steel design software"
authors = ["Osdag Team"]
channels = ["conda-forge", "osdag"]
platforms = ["win-64"]

[dependencies]
osdag = "2025.01.a.2"
python = "3.12.10*"
numpy = "2.2.5"
openpyxl = "3.1.5"
pandas = "2.2.3"
pynput = "1.8.1"
pyqt = "5.15.10"
pythonocc-core = "7.8.1.1"
pylatex = "1.4.2"
pygithub = "2.6.1"
pyyaml = "6.0.2"
smesh = "9.9.0.0"
tbb = "2022.1.0"

[tasks]
run = "osdag"
```

B.4 Managing Environments, Dependencies, and Tasks

B.4.1 Multiple Environments

- Define environments under `[environments]` (e.g., `dev`, `default`).
- Associate them with feature names like `editable` or `noneditable`.

B.4.2 Dependencies

- Global dependencies go under `[dependencies]`.
- Environment-specific dependencies are added under `[feature.<name>.dependencies]`.
- Ensure compatibility by specifying exact or limited dependency versions. [python = ”¿3.10,*”, ”¡3.11.*”]

B.4.3 Defining and Executing Tasks

- Tasks are defined in `[feature.<name>.tasks]` and executed using:
  - `pixi run <task-name>`
- For example:
  - `pixi run osdag-editable` — runs Osdag in editable mode.
  - `pixi run install-editable` — installs the project in editable mode.
  - `pixi run osdag` — runs Osdag in non-editable mode.
- Tasks change depending on the active environment.
- Note:
  - `pixi install` - installs osdag in non-editable mode without launching it
  - running `pixi run osdag` and `pixi run osdag-editable` commands before installing the respective environments will automatically install the respective environments and launch osdag in non-editable or editable mode based on the command executed

Changing and Running Environments

To switch between environments:

- Activate editable environment: `pixi shell -e dev`
- Activate noneditable environment: `pixi shell -e default`

Updating Dependencies

- Add dependencies: `pixi add <package-name>`
- Remove dependencies manually from `pixi.toml`, then run:
- `pixi install` — sync changes from `pixi.toml`.
- `pixi update` — update existing packages to latest compatible versions.

Reproducibility with Lock File

- Pixi generates `pixi.lock` to lock all versions.
- This ensures reproducible builds across systems.

Appendix C

Osdag Latex Environment(Steve's Work)

C.1 Introduction

LaTeX, a widely used document preparation system, typically requires a full TeX distribution like TeX Live or MiKTeX. However, these distributions are large, take time to install, and may require administrative privileges. To overcome these challenges and ensure a portable, lightweight, and platform-specific alternative, the osdag-latex-env was introduced in the Osdag ecosystem.

Here, we outline the structure, purpose, advantages, and functioning of the osdag-latex-env folder, which acts as a self-contained LaTeX environment to compile .tex files without requiring a system-wide TeX distribution.

C.2 Purpose of osdag-latex-env

The primary goal of osdag-latex-env is to:

- Provide a minimal, portable LaTeX toolchain to users of Osdag without requiring external dependencies.
- Enable headless PDF generation of LaTeX documents from within Osdag's installation directory.
- Ensure that the correct versions of all required packages are available in a controlled and pre-tested environment.

  This folder contains everything needed to compile LaTeX documents specifically used by Osdag.

3. Folder Structure The folder typically includes:

```
bin
└─windows.
```

```
│
│     ├── pdflatex.exe.
│     └── other essential binaries
├── texmf-config
│   ├── .configuration files - specific to user
│   └── ls-R
├── texmf-dist
│   ├── bibtex
│   ├── dvipdfmx
│   ├── dvips
│   ├── fonts
│   ├── makeindex
│   ├── metafont
│   ├── mft
│   ├── scripts
│   ├── tex
│   ├── texconfig
│   └── web2c
├── texmf-local
└── texmf-var
```

Appendix D

How to add Osdag Plugins in Installer

D.1 How to add plugin installation in the installer

Plugins are standalone modules that uses osdag to perform calculations and are independent of osdag. Once a plugin is been created and tested plae the module folder in the pixi installer folder (see Chapter 2)

```
Pixi-installer
  └─ version25.01.0.2-win
       ├─ Osdag-2025.01.0.2-win.nsi
       └─ Osdag-v2025.01.0.2-win.exe
├─ osdag-pixi
│   ├─ .pixi
│   │   └─ envs
│   │       └─ default
│   ├─ pixi.exe
│   ├─ pixi.lock
│   └─ pixi.toml
├─ license.txt
├─ Osdag.ico
├─ Osdag_App_icon.ico
├─ Osdag_header.bmp
├─ run.bat
├─ update.bat
├─ Plugin A
├─ Plugin B
└─ Plugin C
```

In the NSIS Script create a new optional section for each plugin.

```
; optional plugins installattion
Section "PluginA" SEC_PluginA
```

```
    SetOutPath "$INSTDIR\Osdag\.pixi\envs\default\Lib\site-packages\osdag\plugins"
    File /r "..\pluginA"
SectionEnd

Section "PluginB" SEC_PluginB
    SetOutPath "$INSTDIR\Osdag\.pixi\envs\default\Lib\site-packages\osdag\plugins"
    File /r "..\pluginB"
SectionEnd
```

Listing D.1: Plugin Sections

The Location, where to copy these plugins are yet to be confirmed and how the plugins will interact with osdag is a question to be discussed.

Appendix E

Nullsoft Scriptable Install System(NSIS)

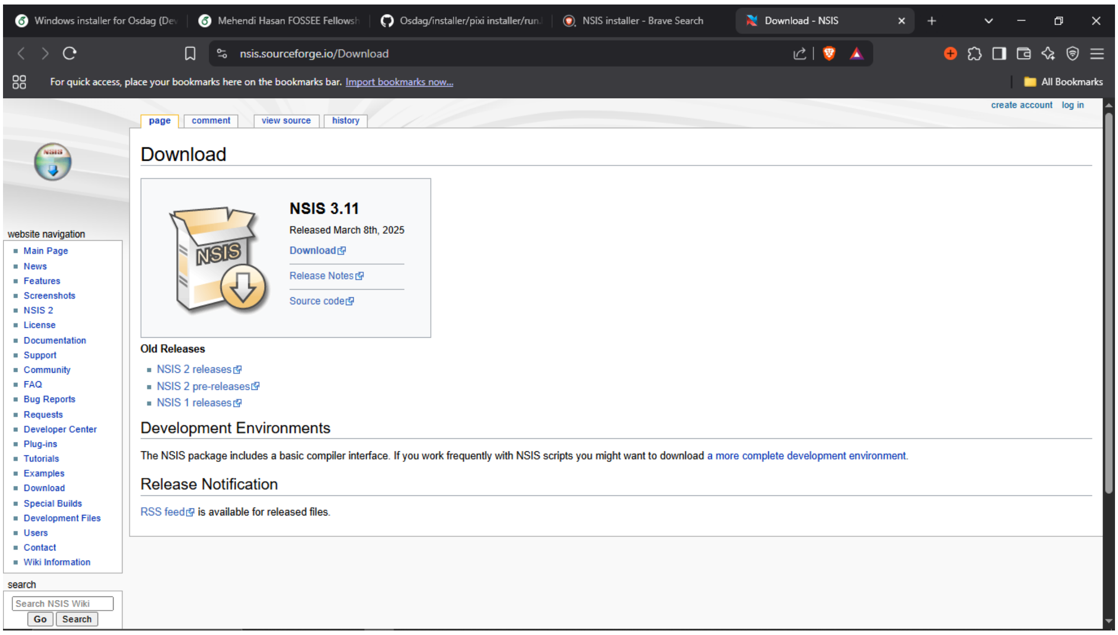
E.1 Introduction

The Nullsoft Scriptable Install System (NSIS) is a professional open-source system for creating Windows installers. Designed to be small, efficient, and flexible, NSIS allows developers to build fully customizable installation packages for their software applications. Its script-driven approach provides powerful capabilities, including support for custom dialogs, logic-based flow control, plug-ins, and extensive file operations. Due to its lightweight nature and broad compatibility with all modern Windows versions, NSIS is widely used for both open-source and commercial software distributions. Whether you're creating simple one-click installers or complex deployment routines, NSIS offers the tools and extensibility to meet your packaging needs effectively.

E.2 Installation

E.2.1 Step 1: Download NSIS

Visit the official NSIS website: NSIS Click on the latest NSIS Installer


Windows installer for Osdag (De
Mehendi Hasan FOSSEE Fellowsh
Osdag/installer/pixi installer/run
NSIS installer - Brave Search
Download - NSIS
nsis.sourceforge.io/Download
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
All Bookmarks

create account log in

page comment view source history

Download

NSIS 3.11

Released March 8th, 2025

Download

Release Notes

Source code

Old Releases

- NSIS 2 releases
- NSIS 2 pre-releases
- NSIS 1 releases

Development Environments

The NSIS package includes a basic compiler interface. If you work frequently with NSIS scripts you might want to download a more complete development environment.

Release Notification

RSS feed is available for released files.

website navigation

- Main Page
- News
- Features
- Screenshots
- NSIS 2
- License
- Documentation
- Support
- Community
- FAQ
- Bug Reports
- Requests
- Developer Center
- Plug-ins
- Tutorials
- Examples
- Download
- Special Builds
- Development Files
- Users
- Contact
- Wiki Information

search

Search NSIS Wiki

Go Search

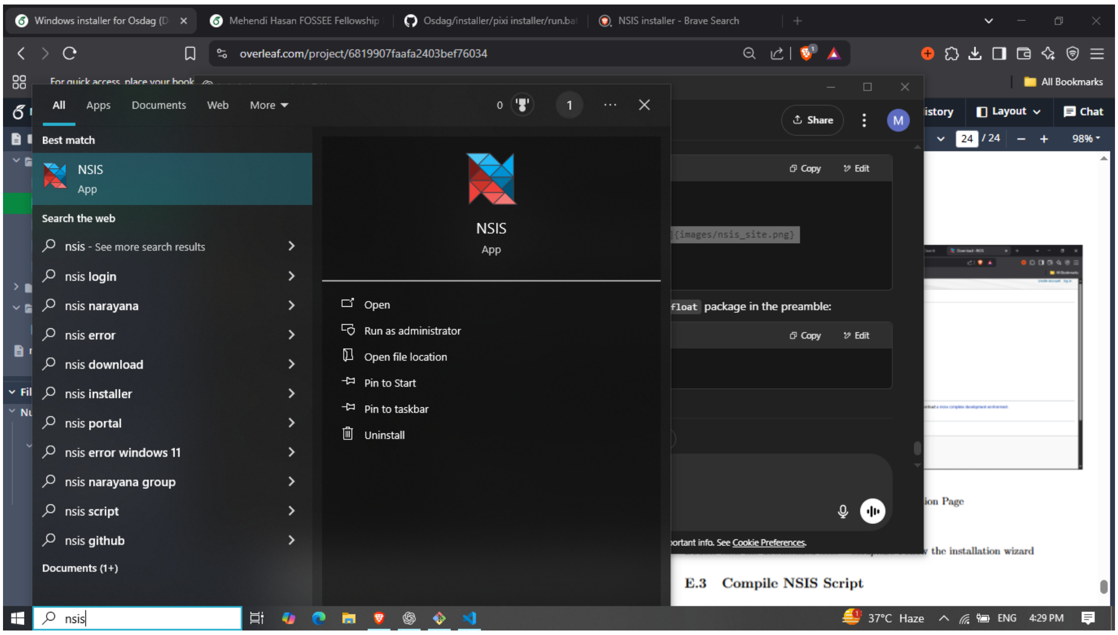
Figure E.1: NSIS Installation Page

E.2.2 Step 2: Run the Installer

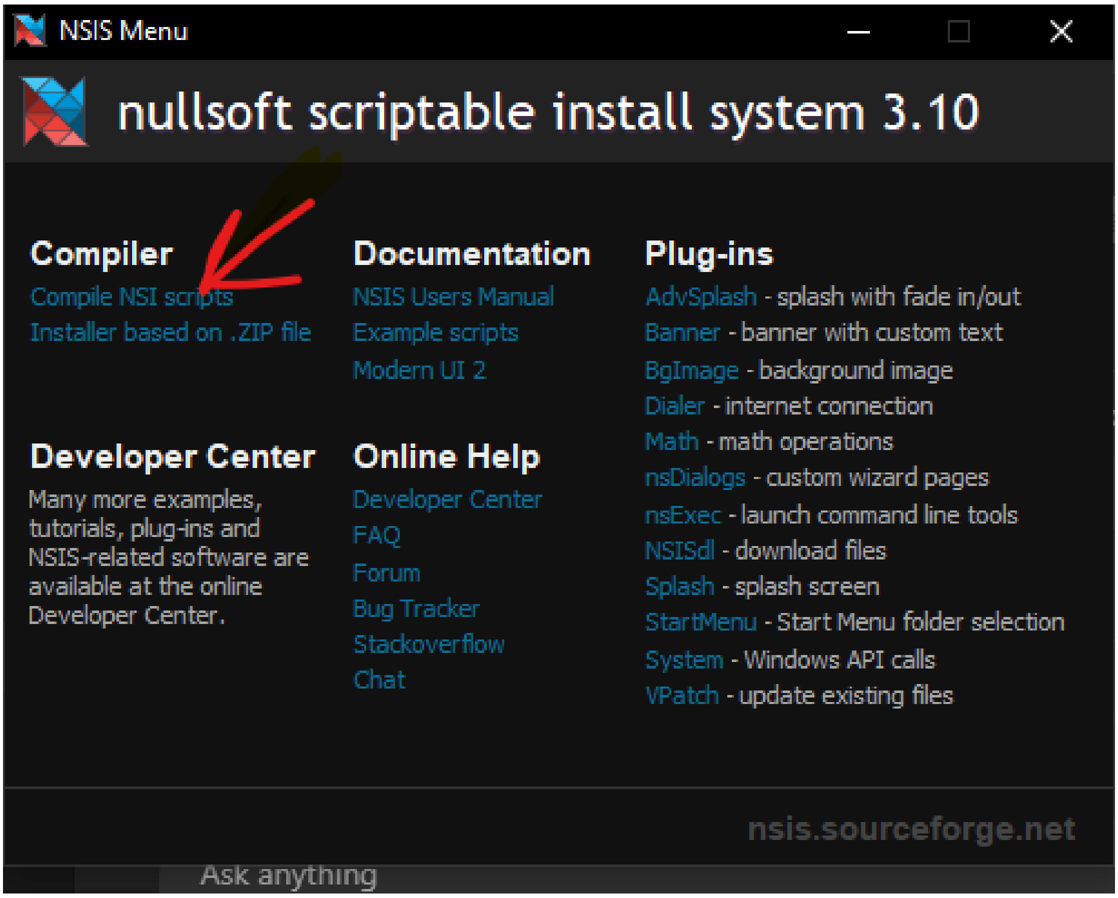
Double-click the downloaded $nsis - setup.exe$ Follow the installation wizard

E.3 Compile NSIS Script

E.3.1 Open NSIS GUI Application



E.3.2 Click Compile NSIS Script



NSIS Menu

nullsoft scriptable install system 3.10

Compiler

Compile NSI scripts

Installer based on .ZIP file

Documentation

NSIS Users Manual

Example scripts

Modern UI 2

Plug-ins

AdvSplash - splash with fade in/out

Banner - banner with custom text

BgImage - background image

Dialer - internet connection

Math - math operations

nsDialogs - custom wizard pages

nsExec - launch command line tools

NSISdl - download files

Splash - splash screen

StartMenu - Start Menu folder selection

System - Windows API calls

VPatch - update existing files

Developer Center

Many more examples, tutorials, plug-ins and NSIS-related software are available at the online Developer Center.

Online Help

Developer Center

FAQ

Forum

Bug Tracker

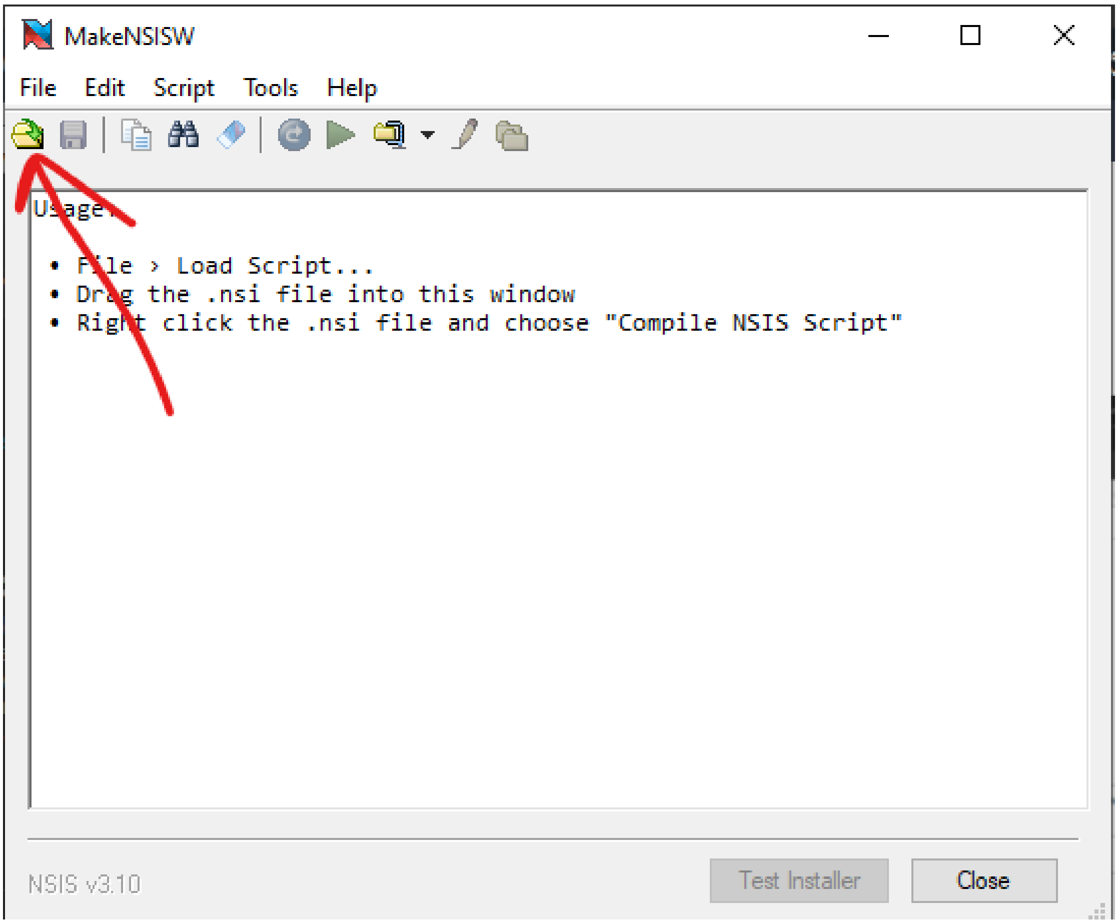
Stackoverflow

Chat

nsis.sourceforge.net

Ask anything

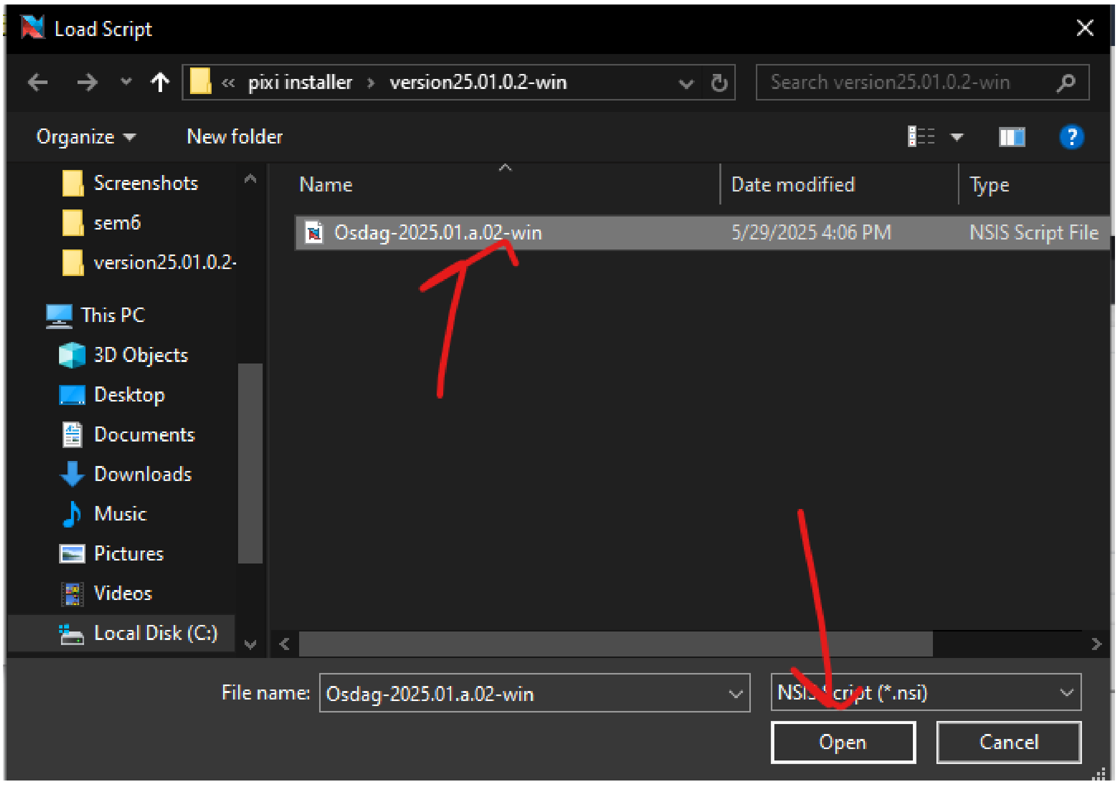
E.3.3 Locate NSIS Script



```
Usage:

• File › Load Script...
• Drag the .nsi file into this window
• Right click the .nsi file and choose "Compile NSIS Script"
```

E.3.4 Select File



Name	Date modified	Type
Osdag-2025.01.a.02-win	5/29/2025 4:06 PM	NSIS Script File

File name: Osdag-2025.01.a.02-win

NSIS Script (*.nsi)

Wait for the compile to complete. It will take time. On completion it will generate a .exe file at the same location where nsis script is located

Appendix F

Test Suite with Installer

Bibliography

[1] Siddhartha Ghosh, Danish Ansari, Ajmal Babu Mahasrankintakam, Dharma Teja Nuli, Reshma Konjari, M. Swathi, and Subhrajit Dutta. Osdag: A Software for Structural Steel Design Using IS 800:2007. In Sondipon Adhikari, Anjan Dutta, and Satyabrata Choudhury, editors, *Advances in Structural Technologies*, volume 81 of *Lecture Notes in Civil Engineering*, pages 219–231, Singapore, 2021. Springer Singapore.

[2] FOSSEE Project. FOSSEE News - January 2018, vol 1 issue 3. Accessed: 2024-12-05.

[3] FOSSEE Project. Osdag website. Accessed: 2024-12-05.