



IT Review and Requirements

How is OpenBCA Downloaded?

OpenBCA is an open-source application that can be downloaded directly from the associated open-source GitHub repository. The repository can be found at [OpenBCA GitHub](#). See the README and accompanying documentation for additional information.

▼ Assets 6

 openbca-app-macos-arm64.zip
 openbca-app-win-x86_64.zip
 OpenBCA Configuration.xlsm
 OpenBCA_Program_Input.xlsm
 Source code (zip)
 Source code (tar.gz)

OpenBCA is distributed as a downloadable .zip package. Once extracted, the package includes the OpenBCA executable and a supporting internal folder containing the application source code, Python components, and required dependencies.

Separate release packages are currently available for macOS and Windows.

Privacy and Data Security

OpenBCA is designed to run entirely on a user's local hardware without requiring an internet connection. As a result, sensitive utility, customer, and planning data remain within the user's environment and are not transmitted to external servers.

Required IT Permissions

Some enterprise environments may block downloaded executables, unsigned applications, command-line tools, or locally bundled Python applications. Before using OpenBCA, IT may need to review and approve:

- **Download and installation access** — permission to download, extract, and install or store the OpenBCA release package in an approved local directory.

- **Application execution** — permission to run the OpenBCA executable, desktop app, or app bundle, including if it is flagged as downloaded, unsigned, or from an external source.
- **Local Python and command-line components** — permission for OpenBCA to run locally bundled Python components, dependencies, or background command-line processes, if applicable.
- **Local file access** — permission for OpenBCA to read input templates from the local working directory and write validation results, model outputs, and local database files to the output folder.
- **Endpoint protection or firewall exceptions** — review of Windows Defender, endpoint protection, firewall, or application-control rules that may block OpenBCA, with exceptions created if needed.
- **Source code and dependency review** — review of the GitHub repository, Technical README, license file, dependency list, and release package, if required by organizational policy.

Minimum System Requirements

Requirement	Recommended / Required
Operating system	Windows and/or macOS, depending on the release package provided
RAM	16 GB recommended; smaller analyses may run with less
Storage	Sufficient local storage for application files, Excel inputs, and generated output database
Processor	Modern multi-core processor recommended
Internet access	Not required for standalone execution after installation/download, unless pulling source code or dependencies
Excel and Macros	Required for preparing/editing input templates

How is OpenBCA Architected?

OpenBCA is built using Python and SQL. It relies on several commonly used Python and data-processing packages, including but not limited to:

- **uv** for Python package management.
- **Pandas ExcelFile** for parsing data from Excel input templates.
- **SQLmesh** to orchestrate data and computational pipelines.
- **DuckDB** for local database management and execution of SQL queries.
- **Streamlit** for the base framework of the user interface.

OpenBCA uses DuckDB to interact with a local .db file created as part of the SQLmesh pipeline. DuckDB executes the SQL queries orchestrated by SQLmesh, which contains the core OpenBCA model logic. For a complete list of dependencies, reviewers can refer to the pyproject.toml file.

The main elements of the OpenBCA framework are:

1. A SQLmesh pipeline (see “excel_input_parsing” folder) parses Excel input templates to generate tables in the OpenBCA input format. The parsing utilizes the [Pandas](#) library, in particular the [ExcelFile](#) class.
2. A SQLmesh pipeline (see “core” folder) runs the SQL queries that make up the OpenBCA computational model.
3. The “user_interface” folder contains the files used to support the OpenBCA interface.
4. The “demo_input_files” folder includes functional sample input files to help users get started.
5. Additional files and folders support the technical README.md, configure file paths, define environment variables, and enable normal application operations.

The Makefile includes command-line instructions for testing functionality and launching the OpenBCA application. When a user double-clicks the OpenBCA executable, the application opens a terminal and runs a Python script. That script launches a local web server, which is the standard operating model for the Streamlit application. The executable also opens a PyWebView container, allowing the Streamlit app to be displayed in a desktop application window.

How is OpenBCA Run?

OpenBCA runs entirely on the user’s local machine and supports an end-to-end workflow without requiring users to write code, use terminal applications, or manage databases. Users prepare inputs in Excel templates, upload the completed files through the OpenBCA interface, launch the model, review results, and download outputs from the application.

Name	Date modified	Type	Size
Today			
 openbca-app.exe	7/15/2026 6:40 PM	Application	29,284 KB
 _internal	7/15/2026 3:40 PM	File folder	

How is OpenBCA Licensed?

OpenBCA is licensed under a BSD-3-Clause open-source license by the Regents of the University of California, through Lawrence Berkeley National Laboratory. Users should review the LICENSE file in the OpenBCA public repository for full terms, including redistribution, modification, attribution, and disclaimer provisions.