

iMS Software Development Kit

Overview (imslib v2.0.x)

Released: 10/06/26

Last Updated: N/A

Overview

The Isomet iMS Software Development Kit (SDK) is an open-source platform for developing, integrating, and operating applications with Isomet digital synthesiser hardware. The SDK includes native C++ libraries for direct device control, Python access for higher-level development, and Windows applications such as iMS HW Server (gRPC) and iMS Studio for communication, configuration, testing, and operation.

Compatibility

	Windows 10+	Ubuntu 24.04
C++ SDK	✓	✓
Python	✓	✓
.NET	✓	✗
iMS Studio	✓	✗
iMS HW Server	✓	✗

Contents

1	Foundation of imslib	1
2	Basic Windows installation	1
2.1	Included applications. 1	
2.1.1	iMS Studio	2
2.1.2	iMS HW Server	2
2.1.3	iMS Firmware Upgrade	2
2.1.4	iMS IP Config	2
3	Wrappers	3
3.1	C# (.NET)	3
3.2	Python	3
4	Linux and embedded systems	3
5	Open-source availability. 3	

ISOMET CORP, 10342 Battleview Parkway, Manassas, VA 20109, USA.

Tel: (703) 321 8301, e-mail: isomet@isomet.com

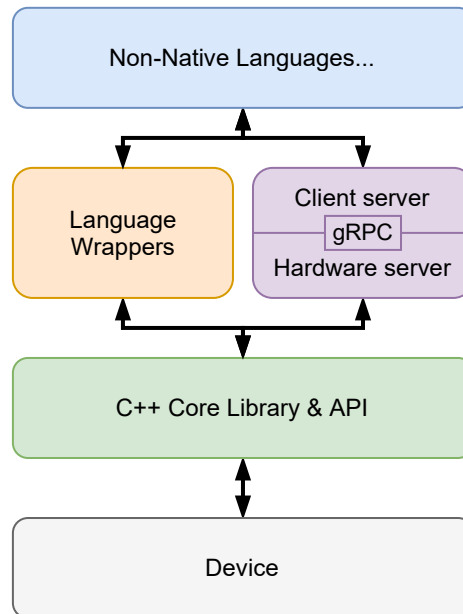
ISOMET (UK) Ltd, 18 Llantarnam Park, Cwmbran, Torfaen, NP44 3AX, UK.

Tel: +44 1633-872721, e-mail: sales-uk@isomet.com

1 Foundation of imslib

At the core of the iMS Software Development Kit is the C++ iMS library and API, which provides the main interface for discovering devices, establishing communications, configuring settings, and controlling supported Isomet hardware. It serves as the foundation for the wider iMS platform, supporting direct C++ development as well as higher-level language bindings.

This architecture gives flexibility for integrating and operating Isomet hardware, while relying on a common software foundation developed and maintained by Isomet. Higher-level interfaces make advanced capabilities more accessible across different workflows and deployment environments.



2 Basic Windows installation

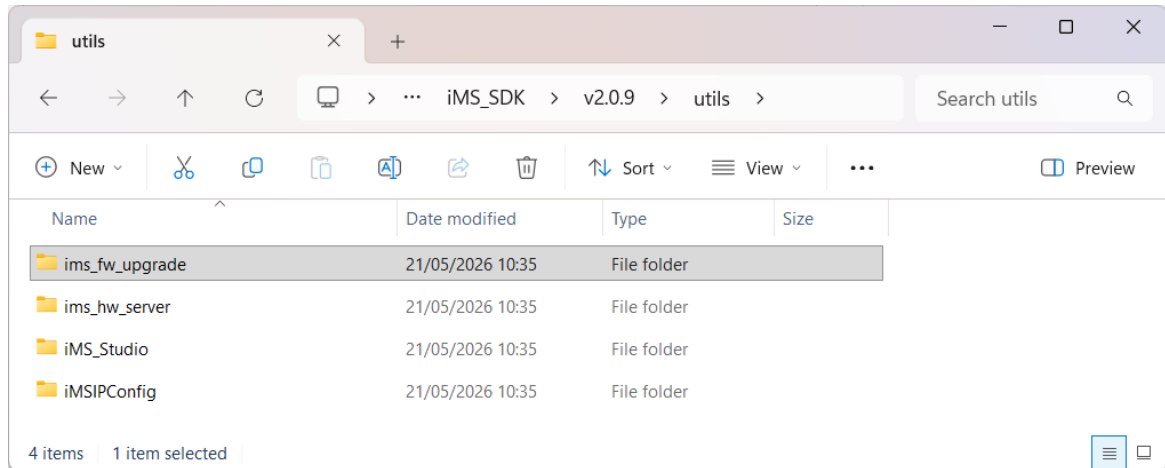
For Windows users, the main iMS SDK installer is the recommended starting point. It installs the core software needed to communicate with supported Isomet iMS hardware and provides the quickest route from installation to device discovery, configuration, testing, and control. For most users, this is the simplest way to begin communicating with an iMS device without first building software from source.

Installers can be downloaded from https://isomet.com/ims4_sw.html. Before beginning the installation process, uninstall any previous version of the iMS SDK and restart the machine. This helps ensure that older components are removed cleanly before the new version is installed.

The installer includes the tools required for common setup and development workflows: iMS Studio, iMS HW Server, iMS Firmware Upgrade, and iMS IP Config. Together, these applications support initial connection, network configuration, firmware maintenance, and interactive operation of the hardware.

2.1 Included applications

The following applications are installed as part of the standard Windows package. They are intended to help users bring up hardware quickly, confirm communication, configure device access, and then progress into test, tuning, development, or integration work as required.



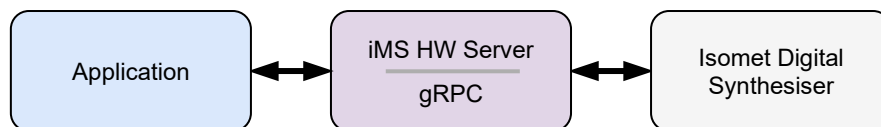
2.1.1 iMS Studio

iMS Studio is the main Windows graphical application for interactive use of iMS hardware. It is primarily intended for test, tuning, and development activities, allowing users to exercise device functions, verify behaviour, create and play operating data, and explore hardware capabilities without writing code first.

App Notes:

- [AN230606 Isomet Studio Software Guide.pdf](#)
- [AN250323 Isomet Studio - Running Image mode example.pdf](#)
- [AN250414 Isomet Studio - Running Tone Buffer example.pdf](#)
- [AN250415 Isomet Studio - Enhanced Tone Buffer example.pdf](#)

2.1.2 iMS HW Server



iMS HW Server provides the communication service used by higher-level applications to access iMS devices. It acts as the hardware communication layer between client software and the connected device and is interfaced with using gRPC. This allows applications such as iMS Studio and other client tools to communicate consistently with the hardware.

2.1.3 iMS Firmware Upgrade

iMS Firmware Upgrade is used to update supported iMS hardware with approved firmware releases. Keeping firmware current helps ensure compatibility with the installed SDK tools and provides access to the latest supported device behaviour and fixes. After installation, the firmware upgrade utility can typically be found in the SDK utilities folder, for example:

```
C:\...\install path...\iMS_SDK\v2.0.x\utils\ims_fw_upgrade
```

App Notes:

- [iMS4 Tools \(FW-Upgrades Logs\)-AN210108.pdf](#)

2.1.4 iMS IP Config

iMS IP Config is used to configure the IP address settings of an iMS controller. This is useful when connecting over Ethernet, changing a device from its default network settings, or preparing hardware for use on a customer network.

3 Wrappers

The iMS SDK exposes the core C++ functionality through language wrappers so that application developers can choose the most appropriate integration layer for their project. The public repositories currently include wrappers for C#/.NET and Python, both of which are built around the shared C++ core library. These wrappers are intended to simplify application development while preserving access to device discovery, configuration, control, diagnostics, signal generation, playback, and related hardware functions.

3.1 C# (.NET)

The C# wrapper is provided by the `imslib-csharp` repository. It builds a C# .NET library around the C++ `ims-lib` module, allowing Windows applications to call into the iMS core library from managed code. The repository uses SWIG interface files and CMake-based build support, with source, test, and build-script folders included for development and validation. This option is most suitable where the end application is a Windows desktop, test, automation, or production tool written in C#.

3.2 Python

The Python wrapper is provided by the `imslib-python` repository and published as the `imslib` package. It provides a higher-level Python API for controlling Isomet acousto-optic synthesizers, including generation of test tones, images, and sequences; downloading and playing data on supported hardware; power control; signal compensation; diagnostics; and configuration. The package supports Python 3.8 or newer. On Windows, users can install the package directly from PyPI using the command `pip install imslib`.

On Ubuntu 24.04, a precompiled wheel is available from the Isomet software download page and can be installed into a Python virtual environment.

4 Linux and embedded systems

The iMS SDK can also be used on Linux and embedded-style systems where a lighter deployment is required or where applications need to run without the Windows desktop tools. Ubuntu 24.04 is the supported Linux target for prebuilt Python wheels. Linux users should create a Python virtual environment, install the supplied wheel, and then run the provided test scripts to verify communication with connected hardware.

Where no prebuilt package is available for the target platform, the Python wrapper can be built locally from the repository using the source tree and its submodules. This route is intended for unsupported platforms or custom embedded deployments and may require the local build tools used by the project, including CMake, Conan, SWIG, and a suitable C++ toolchain. For most standard Windows and Ubuntu 24.04 installations, the prebuilt packages are preferred because they reduce setup time and avoid unnecessary local compilation.

5 Open-source availability

For advanced users, the full raw SDK source is available as open source through the Isomet Corporation GitHub organisation. This provides access to the underlying repositories used to build and maintain the iMS software stack, including the core library and supported language wrappers. Developers who need to inspect the implementation, build from source, integrate the SDK into custom applications, or contribute to platform-specific workflows can use the open-source repositories as the reference starting point.

<https://github.com/Isomet-Corporation>

--- END OF DOCUMENT ---