

# IDA Basics

## Recognition Deployment Options

### OVERVIEW

IDA Recognition is the **core feature for data capture** in PLANET AI's IDA Suite. Despite scenarios like distorted, poor-quality scans with machine-print and difficult-to-read handwriting, IDA Recognition delivers OCR and ICR results with exceptional accuracy. Please refer to the respective [feature datasheet](#) for details.

IDA Recognition can be utilized as a **client-server application** or integrated as a **software development kit (SDK)**. This datasheet outlines the system requirements and specifications for each possibility.

### IDA SERVER

IDA Recognition as part of the IDA Server serves as the standard deployment option. The IDA Server provides a browser interface and can be enhanced with additional IDA features, such as [Classification](#) and [Extraction](#).

IDA utilizes a **gRPC API** to enable communication between the IDA Server and the client. Additionally, we offer two Java SDKs for client-server communication.

#### Supported operating systems

##### *For 64-bit systems*

**Linux:** Ubuntu 18.04 - 23.10, Debian 11, 12; CentOS 8, Red Hat 8.x, 9; LEAP 15.x, SLES 15 SP 4-5

**Windows:** 10, 11

**Windows Server:** 2016, 2019, 2022

**Docker**

#### Additional system requirements:

- At least 12 GB hard disk storage
- At least 16 GB RAM
- CPU-only mode possible

Please contact us for more details on hardware recommendations and sample calculations for document throughput.

### IDA RECOGNITION SDK

For those looking to integrate IDA Recognition directly into software applications, we provide a tailored software development kit (SDK).

**Please note** that the Recognition SDK, by default, decodes based on pre-set language models due to its pre-configured workflow. The .tiff output format is not supported by the Recognition SDK.

#### Additional system requirements:

- At least 6 GB hard disk storage
- **Supported programming languages:** Java, C# (wrapper, provided as a DLL)
- **Separate SDK versions:** CPU only and full CPU/GPU support