

ESENSE SDK

MINDFIELD ESENSE SDK ZUR ENTWICKLUNG EIGENER MACOS-SOFTWARE

[zur Produktseite im Shop](#)
Artikelnummer: SDK-ESE-MAC
Preis: €249,00

MINDFIELD ESENSE SDK ZUR ENTWICKLUNG EIGENER MOBILER APPS (IOS/ANDROID)

[zur Produktseite im Shop](#)
Artikelnummer: SDK-ESE-MOB
Preis: €399,00

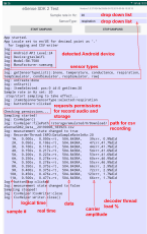
MINDFIELD ESENSE SDK ZUR ENTWICKLUNG EIGENER WINDOWS SOFTWARE

[zur Produktseite im Shop](#)
Artikelnummer: SDK-ESE-WIN
Preis: €249,00

Use the IntelLabs MacResponse SDK

Technical Prerequisites: The SDK is compiled with Microsoft .NET Framework 4.5.2. Take a look at the [MacResponse readme](#), it contains a fully working example of the SDK.

- Reference the Class Library „[IntelLabsMacResponse.dll](#)“ in your project
- Import Namespace IntelLabsMacResponse
`using IntelLabsMacResponse;`
- Create new instance of class MacResponseService
`MacResponseService mrs;`
- Create new instance of MacResponseClientEvent (needed for value filtering, see setting usage algorithms). For MacOs value calculations, use the targetPlatform parameter in the constructor
`MacResponseClientEvent targetPlatform;`
- Subscribe to MacResponseService class callback event. The event is fired multiple times a second and contains the current read value. Add the value to the MacResponseClientEvent by value filtering, calibration, and getting MacOs file value
`MacResponseService macResponseService = new MacResponseService();`
- Start MacResponseService
`macResponseService.Start();`
- When done, either stop or dispose MacResponseService
`macResponseService.Dispose();`



Use the IntelLabs MacResponse SDK

Technical Prerequisites: The SDK is compiled with Microsoft .NET Framework 4.5.2. Take a look at the [MacResponse readme](#), it contains a fully working example of the SDK.

- Reference the Class Library „[IntelLabsMacResponse.dll](#)“ in your project
- Import Namespace IntelLabsMacResponse
`using IntelLabsMacResponse;`
- Create new instance of class MacResponseService
`MacResponseService mrs;`
- Create new instance of MacResponseClientEvent (needed for value filtering, see setting usage algorithms). For MacOs value calculations, use the targetPlatform parameter in the constructor
`MacResponseClientEvent targetPlatform;`
- Subscribe to MacResponseService class callback event. The event is fired multiple times a second and contains the current read value. Add the value to the MacResponseClientEvent by value filtering, calibration, and getting MacOs file value
`MacResponseService macResponseService = new MacResponseService();`
- Start MacResponseService
`macResponseService.Start();`
- When done, either stop or dispose MacResponseService
`macResponseService.Dispose();`