OriginC Spectral Analysis

Developer Guide

Last Update: 07.03.2022 (v1.2.2)

The *OriginC Spectral Analysis* package can easily be extended in functionality by installing plugin packages to *OriginPro*. The GitHub repository of the project holds an exemplary plugin in the *src* directory. A packaged version is also provided with the releases in the *build* directory.

Plugins should be installable in the same way as the main package. I.e., they should be provided as *OriginPro* packages in the `.opx` format.

Contents

1 File Structure

In principle, the file and folder structure of the plugin project is up to the developer if all references are defined correctly. For packaging, it is however more than useful to have all script files enclosed in a single parent directory named as the plugin itself.

As all source files will be copied to the *User File Folder* during install, not providing a parent directory will create a mess and can espacially lead to problems with ambiguous `include` calls.

2 Referencing

When referencing methods or functions of the main package, the relative path from the plugin directory must be given. E.g., if the plugin resides in its own folder *Example Plugin*, the main project can be included like this:

```
#include "..\Spectral Analysis\headers\lang.h"
```

3 Building

After creating the plugin files, the plugin can be packaged via the *OriginPro Package Manager* in the *Tools* menu. Pay attention to include the whole source file folder and not only the included files to keep the file structure alive. The *Example Plugin* package provided with the main package can be opened as a starting point. In the *after installation* dialog, the install script of the plugin should be executed.

4 Installation

Each plugin should be provided with an installation file, e.g., `install.ogs` in the *Example Plugin*. This script will make the plugin functions available in the *OriginC* workspace after installation.

In the example file, the plugin is supposed to be structured like the main package, with a main script file in the top-level directory and helper files in the *headers* sub-directory. First, all old files contained in the current or previous version of the plugin are removed from the *Code Builder Workspace* to prevent double loading of files and classes. Afterwards, all files contained in the current version of the package are referenced in the *User Autoload* folder and compiled and linked. This procedure makes the plugin methods available at every start of an *OriginPro* instance.