

Developer Guide

This guide describes the Open Java API of MagicDraw or a Cameo Suite product, such as Cameo Systems Modeler, and provides instructions on how to implement custom plugins, add actions to the menus or toolbars, change UML model elements as well as create new patterns. Though MagicDraw is used in examples mainly, other modeling tools, since they are based on the MagicDraw modeling platform, performs the same.

The program core code and used libraries are packaged in jars files that are located in *<modeling tool installation directory>lib* (and its subdirectories).

Some plugins also exposes Open API. Plugins libraries can be located in *<modeling tool installation directory>plugins\<plugin_name>*

The code can be divided into the following three API scope groups:

- **OpenAPI** – a code for public usage, stable through builds and versions.
- **InternalAPI** – a code for the internal NoMagic usage only, may change through builds and versions without any restrictions.
- **NotAPI** – not an API code, it may change in each build.

Java annotations are used to distinguish which API scope in the provided code belongs to. Annotations of the type (class, interface, enum) declaration:

Annotation	API Scope	Scope applies to
@OpenApiAll	OpenAPI	The annotated type declaration and all declared members, constructors.
@OpenApi	OpenAPI	The annotated type declaration only. Each declared member, constructor is annotated explicitly with the @OpenApi or @InternalApi annotation (see the following table).
@InternalApi	InternalAPI	The annotated type declaration and all declared members, constructors.
@NotApi	NotAPI	The annotated type declaration and all declared members, constructors.

Annotations of the member, constructor declaration of the @OpenApi type:

Annotation	API Scope	Scope applies to
@OpenApi	OpenAPI	The annotated declared member or constructor.
@InternalApi	InternalAPI	The annotated declared member or constructor.

Only code from the OpenAPI scope should be used to extend the modeling tool. All the InternalAPI and NotAPI scope code are additionally annotated with the *@Deprecated* annotation. This allows seeing the illegal API usage in all modern IDEs without an additional configuration.

We provide a set of plugin samples in *<modeling tool installation directory>lopenapi/examples*. By using these examples, you can find out on how to use an OpenAPI.

In the generated [javadoc](#), you will find the detailed description of classes, their attributes, and operations. Javadoc is located in *<modeling tool installation directory>lopenapi/docs*.