Plugin descriptor

A plugin descriptor is a file written in XML and named *plugin.xml*. Each descriptor contains properties of the one plugin. The descriptor file should contain the definitions of elements of the one plugin.
The *plugin.xml* file consists of the following elements:

- plugin
- requires
- api
- required-plugin
- runtime
- library
- help

The following tables describe them in more details.

plugin

Attribute	Description
id	A plugin ID should be unique. It is used to identify a plugin by a plugin's manager internals and by requirements of other plugins. For example, *my.first.plugin.0.*
name	A plugin name. There is no strict rules applied to this attribute. For example, *Example plugin*.
version	A plugin version. The version can be used to check other plugins dependencies, if internalVersion is not defined.
internalVersion	A plugin internal version. It is used to check other plugins dependencies.
provider-name	A plugin provider name, which is a company or an author name. For example: *No Magic*.
class	A full qualified class name. The class must be derived from *com.nomagic.magicdraw. plugins*. A plugin and stored in plugin runtime library. This class will be loaded and initialized by the plugin's manager. For example, *myplugin.MyPlugin*.
ownClassloader	Optional; the default value is *false*. Indicates, if to use an own plugin's classloader (separate from other plugins). All program plugins are loaded by the **one** classloader. If there are plugins that cannot be loaded by the same classloader (for example, because of conflicts in plugin libraries versions or other), their descriptors must be defined to use own class-loaders.
class-lookup	Optional; possible values are *LocalFirst*, *GlobalFirst*, the default value is *GlobalFirst*. Specifies the priority of a "parent" class loader, if a plugin uses ownClassloader. LocalFirst forces to load classes from the plugin class loader even if such classes exist in the modeling tool core class path. This option should be used, if, in your plugin, you want to use different versions of same libraries that are used in the core.

Nested elements

Element name	Description
requires	A modeling tool API version required by the plugin. Plugins and their versions required by the plugin.
runtime	Runtime libraries needed by a plugin.

requires
Nested elements

Element name	Description
api	A required modeling tool API version.
required-plugin	Plugin(s) that are required to run the plugin.

api

Attribute	Description
version	A required modeling tool API version. For example, *1.0.*

required-plugin

Attribute	Description
id	The ID of a required plugin. For example, *my.first.plugin.0.*
name	A name of a required plugin.
internalVersion	An internal version of a required plugin. If it is not defined, the "version" is used to check if the required plugin is suitable.
version	A version of a required plugin. If it is not defined, any required plugin version is suitable. For example: *1.1.*

runtime

Nested elements

Element name	Description
library	A runtime library for a running plugin.

library

Attribute	Description
name	A name of the required library. For example, "patterns.jar".

help

Java Help support will be deprecated in 18.3 and up versions, so we do not recommend using it anymore.

Attribute	Description
name	A name of a compressed JavaHelp file (*.jar* file). ⓘ Adobe RoboHelp provides support for the JavaHelp format and automatically creates all Java-based Help features and the HTML-based features such as an HTML content and hypertext links.
path	A relative path to the JavaHelp file.