

Archived Plugin Development / Open API

1. Login with appropriate right, preferable 'Administrator'.
2. Remove MD_customization_for_SysML.mdzip from your Teamwork Server.
3. Create a new SysML project.
4. Add the newly-created SysML project to Teamwork.
5. In the Add Project to Teamwork Settings dialog, select the Local Modules tab.
6. Ensure that the Action for MD_customization_for_SysML.mdzip is Add to Teamwork.
7. Click the Add button in the Add Project to Teamwork Settings dialog.

MD_customization_for_SysML.mdzip will be automatically updated in your Teamwork.

DoDAF Plugin 2.0 is not compatible with DoDAF Plugin 1.0. Projects, created with DoDAF Plugin 1.0, cannot be opened with DoDAF Plugin 2.0 and vice versa.

Resource Manager needs more information which is stored in resource descriptor. It must know what files were installed because it must be able to perform uninstall operation.

I have attached our free form custom diagram. It would be best if you will look through this sample and I'll describe main rules for descriptor writing. You will find descriptor file in "data/resourcemanager/MDR_Custom_Diagram_Free_Form_1513_descriptor.xml".

- xml tag

- "id" is most important. It must be unique. To prevent duplicates in future use "22**", for example "2201".

- "name" must be unique between all MagicDraw resources.

- "type" must be one of: "Custom Diagram", "Plugin", "Profile", "Sample", "Template".

- is your resource version

- will be needed only if your resource depends on other resources.

- defines files which will be installed. Do not use "*" (all files in directory) in common directories like " /samples" because uninstalling your resource will remove all files from this directory!

- each tag defines one file which must be copied to "MagicDraw root" directory.

Descriptor file must be placed in your resource zip file in directory "data/resourcemanager" with specific naming. Name of descriptor file is formed from values from tag: "MDR_..._descriptor.xml", all spaces in names is replaced by underscore symbol "_".

[Resource sample you may find here.](#)

1. Create Java Project.
2. Include all *.jar files from "MagicDrawLib" folder into Project Build Path. Exclude "md_commontw_api.jar" and "md_commontw.jar".
3. Write plug-in. Start from main Plug-in class by extending "com.nomagic.magicdraw.plugins.Plugin".
4. Create Plug-in descriptor file "plugin.xml".
5. Make clean plug-in directory in MagicDraw plug-ins directory (./plugins/), copy "plugin.xml" to it. Do not copy any binaries to MagicDraw plug-ins directory or they will have conflicts with your binaries in Eclipse.
6. Create new Debug configuration to run Application.
7. select "Project" your plug-in project.
8. search for Main class, select "com.nomagic.magicdraw.Main".
9. add "-Xmx600M -XX:PermSize=40M -XX:MaxPermSize=150M" to Arguments.
10. Apply changes.

More details can be found in "[MagicDraw Open API User Guide](#)"

Hints:

- don't forget to include all sub folders from "MagicDrawLib" into your Build Path;

- create jar file from Eclipse: use "Export / JAR file";

- you wouldn't have to copy files each time if you will select destination jar file already in "MagicDraw\plugins" folder;

- save jar file description to build jar file later;

You may find full MagicDraw OpenAPI javadoc in *<modeling tool installation directory>\openapi\docs* or download it from http://jdocs.nomagic.com/2022x_Refresh1/.