

Resource dependent plugin

Starting with MagicDraw version 16.6, there is a new functionality to require to load plugins for a particular project. This feature was created to avoid an incorrect data load because of missing plugins. Every plugin can provide a name and version of a plugin to be required for the correct project load. To become a resource dependent plugin, your plugin class must implement the *com.nomagic.magicdraw.plugins.ResourceDependentPlugin* interface.

ResourceDependentPlugin has three special methods:

- a *isPluginRequired(Project)* method is called on saving a project. The plugin must return *true*, if a given project uses resources from the plugin.
- a *getPluginName()* method should return a plugin name.
- a *getPluginVersion()* method should return a version of the plugin.

The following example illustrates an implementation of *ResourceDependentPlugin*:

```
package myplugin;
import ...
public class MyPlugin extends Plugin implements ResourceDependentPlugin
{
    @Override
    public void init()
    {...}
    @Override
    public boolean close()
    {...}
    @Override
    public boolean isSupported()
    {...}

    @Override
    public boolean isPluginRequired(Project project)
    {
        return ProjectUtilities.findAttachedProjectByName(project,
"my_profile_filename") != null;
    }
    @Override
    public String getPluginName()
    {
        return this.getDescriptor().getName();
    }
    @Override
    public String getPluginVersion()
    {
        return this.getDescriptor().getVersion();
    }
}
```

This plugin is required for a project, if the project contains a used project "*my_profile_filename*". The plugin name and version will be saved into the project's XMI file.