

CONCEPTION DE LOGICIELS EXTENSIBLES

Manuel du plugin PluginManager

Brian GOHIER, Yannis GREGO, Kevin MOKILI, Pierre NOYON, Aida TAPSOBA
28 avril 2013

Table des matières

1	Interface	2
2	Fonctionnalités du plugin	2
2.1	Ajouter un plugin	2
2.2	Ajouter un plugin dépendant du plugin : "PluginName"	3
2.3	Importer un plugin	3
2.4	Afficher le fichier de configuration	3
2.5	Actualiser les configurations	3
2.6	Charger le plugin	3
2.7	Supprimer le plugin	3

1 Interface

Le plugin "**PluginManager**" est un plugin actif, il a donc accès aux différentes fonctions du coeur. Ce plugin va afficher tous les plugins présents sur la plate-forme et va pouvoir les modifier et interagir avec eux. Voici à quoi ressemble l'interface de ce plugin :

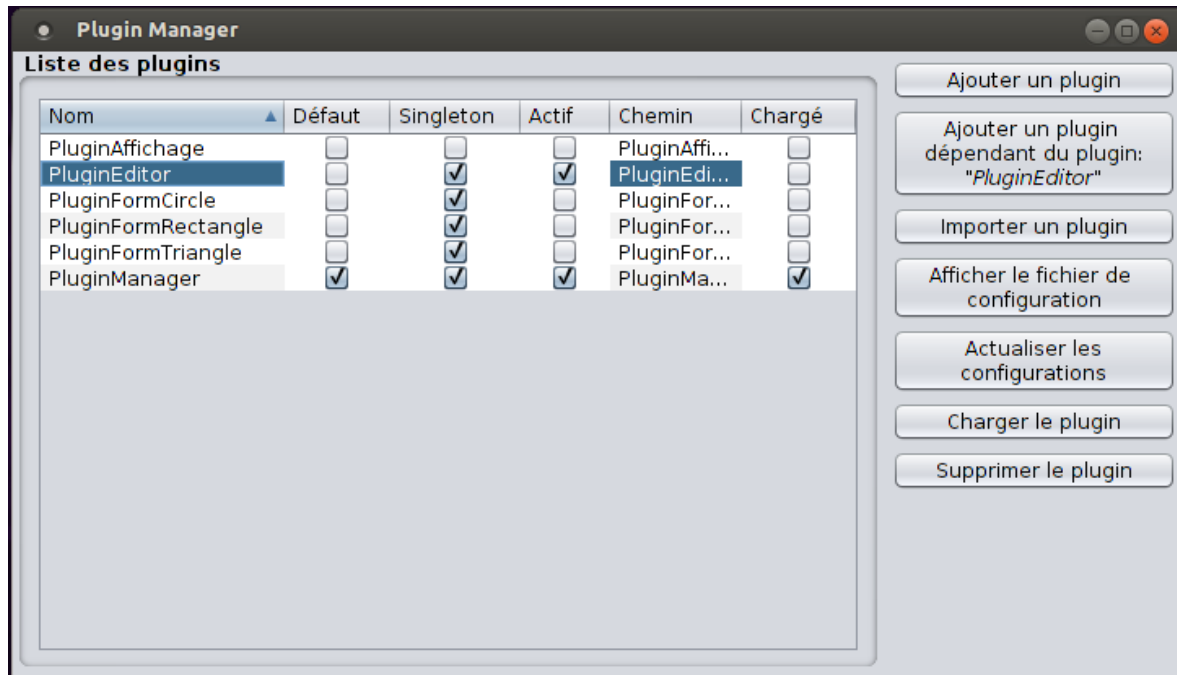


FIGURE 1 – Interface du plugin

On peut voir ici la liste des plugins qu'a trouvé le coeur ainsi que leur état et leurs paramètres. Il est possible de sélectionner un plugin et d'effectuer des modifications sur celui-ci via les *CheckBox* ou les boutons.

2 Fonctionnalités du plugin

Depuis cette interface il est possible de créer, importer, modifier ou même charger des plugins. Nous allons lister toutes les capacités de ce plugin.

2.1 Ajouter un plugin

Le bouton "**Ajouter un plugin**" permet de créer un plugin dans le *workspace* d'eclipse. Il est possible d'y définir le nom du plugin, s'il est actif, s'il doit être chargé par défaut et si c'est un singleton. Quand vous créez un plugin de cette manière, le projet eclipse est créé contenant la classe principale du plugin, par défaut une classe qui porte le nom de ce plugin. Dans cette classe on a un constructeur qui est soit un constructeur par défaut si vous avez défini que ce plugin est passif (pas actif), ou un constructeur prenant en paramètre un objet de type *ICore* s'il est actif, ainsi qu'une fonction main pour la compilation. Le projet est configuré de manière à utiliser la librairie "**core_lib.jar**" qui est copiée depuis celle utilisée par le plugin "**PluginManager**". Un fichier de configuration sera également automatiquement créé.

Il est indiqué, une fois le plugin créé, qu'il est nécessaire de l'importer dans eclipse afin de l'avoir en tant que projet eclipse.

Cette fonction créer donc un plugin tout à fait fonctionnel, il suffit de le compiler pour pouvoir l'importer puis l'utiliser dans la plate-forme.

2.2 Ajouter un plugin dépendant du plugin : "PluginName"

Ce bouton a la même utilité que le précédent. Cependant ici il s'agit de sous-plugin du plugin qui est actuellement sélectionné dans le plugin "**PluginManager**". Ainsi, par exemple pour le plugin "**PluginEditor**", il est possible de créer un sous-plugin qui sera utilisé dans son plugin parent. Une librairie supplémentaire sera créée dans ce projet eclipse. Il s'agit de la librairie qui contient les interfaces dont vont dépendre les sous-plugins. C'est pourquoi il est indispensable de bien indiquer, dans le plugin parent, le bon chemin vers les interfaces nécessaires.

2.3 Importer un plugin

Ce bouton va permettre d'importer les fichiers binaires d'un plugin vers le dossier des plugins de la plate-forme. Une fois votre plugin créé et compilé, avec ce bouton vous n'aurez donc pas à copier manuellement les fichiers binaires vers le dossier de la plateforme. Quand vous cliquez sur ce bouton, une fenêtre s'ouvre (par défaut placée dans le *workspace* d'eclipse) pour vous permettre de choisir quel dossier correspond au projet (projet eclipse) de votre plugin à importer. Sélectionnez votre plugin à importer et validez. Votre plugin sera alors référencé dans la plate-forme ainsi que sur le plugin "**PluginManager**" et vous pourrez donc le modifier ou le charger.

Le fichier de configuration de la plateforme sera alors mis à jour pour répertorier ce plugin.

2.4 Afficher le fichier de configuration

Cette action va vous permettre de visualiser avec votre éditeur de texte par défaut (pour les fichiers *.ini*) le fichier de configuration du plugin sélectionné. Vous pouvez alors le modifier directement dans l'éditeur.

2.5 Actualiser les configurations

Avec ce bouton vous pourrez actualiser les paramètres d'un plugin depuis son fichier de configuration ou son état s'il y a eu du changement.

2.6 Charger le plugin

Ici on va chercher à instancier le plugin sélectionné. On va appeler le chargement de la plate-forme, donc tout est délégué à cette dernière, de manière à ce que si le mode de chargement de la plate-forme est modifié après maintenance par exemple, aucune modification ne soit nécessaire dans ce plugin. On actualisera l'état du plugin en conséquence du résultat du chargement de la plate-forme.

2.7 Supprimer le plugin

Cette suppression va non seulement supprimer le plugin dans la liste des plugins du "**PluginManager**", mais aussi dans la liste que possède la plate-forme, dans son dossier (le dossier est alors complètement supprimé) et son fichier de configuration. Il est donc nécessaire d'utiliser cette fonction avec précaution.