

В общем, появилась идея попытаться порешать эти проблемы и начать с совместного написания статьи о том как собирать и поднимать плагин, о том, какие проблемы при этом могут возникнуть и как их решать. Разработка

[Плагин](#), который генерирует код лексера из описания грамматики в bnf-файле.

[Дока](#) к нему лежит на гитхабе рядом с сорцами.

[Заметки](#) о разработке плагинов (об AST, PSI и всяких нюансах)

[IntelliJ IDEA PSI Cookbook](#)

Сборка haxe-плагина для IntelliJ IDEA

Во-первых, очевидно, что нам потребуется клонировать репозиторий

<https://github.com/JetBrains/intellij-haxe>. В репозитории уже имеется файл проекта для IDEA, но настраивать проект все равно придется.

1. Создать SDK в настройках проекта (Ctrl+Shift+Alt+s) → SDKs → плюсик → IntelliJ Platform Plugin SDK → указать каталог с установленной идеей. Спросит о версии JDK. В руководстве по разработке плагинов рекомендуют 1.6, но у меня с 1.7 дело пошло лучше.
2. В настройках же проекта, но уже на развороте модулей на вкладке Dependencies указать этот SDK для каждого из модулей
3. **Зависимости.** Плагин haxe зависит от флексового плагина, то есть для модуля `intellij-haxe`: + зависимость → каталог/jars → выбрать путь до `%IDEA_PATH%\plugins\flex\lib`

Как правильно заметил Александр, зависимости нужно подключать к SDK, а не к модулю.

4. Для сборки всех плагинов, у в которых есть элементы интерфейса, собранные на **UI Designer** (наш случай) нужно иметь установленный плагин UI Designer. Иначе целевой плагин скомпилируется, но при попытке вызвать соответствующий диалог (вкладка haxe в настройках проекта и т.п.) будет валиться с нуллпоинтером.

Для конечного использования рекомендуется собирать плагин при помощи ant: меню build → build artifacts. В подкаталоге проекта out соберутся бинарники пригодные для установки в идею через "установить плагин с диска".

В принципе, если настройка проекта была выполнена правильно, то по shift+f9 вполне должна запускаться отладка в отдельной копии идеи (у нее свои настройки, можно в ней задать другую конфигурацию плагинов).

5. Для установки собранного плагина штатным образом, нужно создать zip файл, внутри которого будет каталог `haxe`, располагающийся в `$PROJECT_ROOT$/out/artifacts`. Получившийся zip можно устанавливать штатным образом: раздел плагин в окне настроек → установить с диска.

Известные проблемы

1. С последними версиями исходников и идеи на сегодня (23.4.14) есть проблема конфликта с Flex плагином:

~~у Haxe плагина стоит зависимость от флэшовой, и оно при старте идеи с обеими плагинами конфликтуют в общих частях (что то там с переинициализацией уже существующих темплейтов).~~

~~если в целевой идее отключить флэш плагин, то project structure работает норм. понятно, что это не решение, но хотя бы ясно, куда копать.~~

Если зависимости подключены правильно, то проблема не воспроизводится.

2. Не проверял на последних версиях, но при сборке более ранних была следующая проблема: самосборный плагин категорически отказывался показывать окно `messages` (ошибки и ворнинги компиляции, которые позволяют переходить на соответствующее место по `f4`) – ошибки в этом случае вроде бы вываливались просто в консоль; кроме этого плагин не совсем хорошо ловил `exit`-коды компилятора (тестировал с `OpenFL` проектом), и мог просто замереть в ожидании, если компиляция оборвалась с ошибкой.

Ссылки

<http://habrahabr.ru/post/187106/>

<http://confluence.jetbrains.com/display/IDEADEV/PluginDevelopment>

также может оказаться полезным пошариться по форкам официального плагина