

Обзор имеющихся библиотек OneScript

Введение

Для начала разберемся с терминами.

Тем, кто не знает, что такое OneScript, стоит познакомиться с первой вводной статьей от самого автора Как решать повседневные задачи с помощью OneScript (<https://infostart.ru/public/327581/>) и с сайтом проекта OneScript (</redirect.php?url=aHR0cDovL29zY3JpcHQW8v>). На Инфостарте недавно появилось несколько хороших статей на тему "быстрого старта".

Как только скриптописатель выходит за рамки Сообщить("Привет, мир!") и начинает решать прикладные задачи, у него появляется желание найти какие-то готовые куски кода или целиком разработанную функциональность. Большинство полезных скриптов упаковывается в так называемые **пакеты**, упрощающие установку этих скриптов на локальную машину и их последующее использование.

Все пакеты делятся на два типа:

1. **Библиотеки**, подключаемые в скрипты через директиву #Использовать и добавляющие разные полезные классы и модули
2. и **Приложения**, которые помимо реализованной на OneScript логики имеют специальные обертки запуска из командной строки (cmd/sh). Часть приложений помимо запуска из командной строки позволяют себя подключать через #Использовать (т.е. как библиотеки), но это скорее бонус и желание разработчика приложения, чем возможность, на которую всегда можно надеяться.

Все более-менее работающие и известные пакеты собраны в двух местах:

1. Организация **oscript-library** на github.com - <https://github.com/oscript-library> (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeQ==) - содержит оригиналы или форки репозиторий, в которых ведется разработка пакетов. Там лежит весь исходный код, чаще всего там же находится и баг-трекер.
2. Собранные пакеты выложены на специальном **хабе пакетов** <http://hub.oscript.io/download/> (/redirect.php?url=aHR0cDovL2h1Yi5vc2NyaXB0LmlvL2Rvd25sb2FkLw==) - оттуда их можно скачать руками, если по каким-то причинам не работает пакетный менеджер. Страничка поражает своим минимализмом, но и Москве, как говорится, не сразу стало 870 лет.

Менеджер пакетов OneScript - opm

Перед обзором имеющихся пакетов я хочу немного рассказать о самом главном пакете-приложении в экосистеме OneScript - его пакетом менеджере. **opm** (onescript package manager) призван упростить установку, обновление и публикацию пакетов из/в хаб пакетов. Данный пакет входит в "стандартную поставку" движка OneScript и готов к работе с момента установки самого движка. С его помощью за одну команду можно установить любой доступный пакет.

Например, библиотека-помощник работы с Яндекс.Диском ставится с помощью команды **opm install yadisk**. Обновить все установленные пакеты разом можно через команду **opm update -all**, а **opm install -all** разом установит все доступные в хабе пакеты. Практически каждое приложение имеет встроенную справку по командам. Выполнив **opm help** мы получим информацию о всех имеющихся командах, а, например, **opm help install** выдаст справку конкретно по команде установки пакетов.

Пакетный менеджер имеет свой файл настроек, где можно указать, например, параметры подключения через прокси. Из особенностей работы отмечу лишь довольно очевидное ограничение прав. OneScript по умолчанию ставится в каталог C:/Program Files (x86)/OneScript (или в /usr/share/onescript на linux) и без прав администратора ни один пакет не обновится. Решение простое - либо выдаем себе права на каталог установки OneScript, либо запускаем командную строку от имени администратора/суперпользователя.

Библиотеки

Большая часть функциональности заложена в библиотеках. Здесь я перечислю и кратко опишу все библиотеки, доступные в хабе пакетов. Начну с так называемого "набора стандартных библиотек", поставляемого с OneScript, продолжу всеми остальными библиотеками в алфавитном порядке. После названия библиотеки будет идти ссылка на основной репозиторий, в котором ведется разработка. Иногда этот репозиторий будет вести на личный репозиторий автора библиотеки, однако, все перечисленные здесь библиотеки собраны в организации oscript-library на GitHub.com.

Поставляемые библиотеки

1commands - <https://github.com/oscript-library/1commands> (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS8xY29tbWFuZHM=)

Библиотека для запуска различных приложений из OneScript.

Думаю, многие знают про 1сный метод ЗапуститьПриложение() - с его помощью можно запустить любое приложение (и даже открыть файл в ассоциированном приложении). Однако у него есть ряд недостатков:

- Тяжело анализировать консольный вывод такого приложения
- Не всегда удобно склеивать строку для запуска
- Нет управления допустимыми кодами возврата
- Приходится изобретать велосипед для запуска более сложных команд или последовательности команд под win/linux

Данная библиотека добавляет два класса для запуска приложений - Команда (если команда одна) и КомандныйФайл (если команд несколько). Пример запуска Команды из README проекта:

```
Команда = Новый Команда;  
  
Команда.УстановитьКоманду("oscript");  
Команда.ДобавитьПараметр("-version");  
  
// или сразу Команда.УстановитьСтрокуЗапуска("oscript -version");  
  
КодВозврата = Команда.Исполнить();  
Сообщить(КодВозврата);  
  
Сообщить(Команда.ПолучитьВывод());
```

asserts - [https://github.com/oscript-library/asserts \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9hc3NlcnRz\)](https://github.com/oscript-library/asserts (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9hc3NlcnRz))

Реализация функциональности "утверждений" и "ожиданий".

В основном используется в тестах, однако так же позволяет, например, проверять текущее состояние скрипта - заполненность всех нужных параметров и их корректность.

Реализована в виде двух независимых модулей - Утверждения и Ожидаем. Каждый из этих модулей содержит функции проверки переданных значений - на равенство, тип, состав и прочее. Хорошая вводная статья и примеры использования опубликованы в виде статьи на Хабре ([/redirect.php?url=aHR0cHM6Ly9oYWJyYWWhYnIucnUvcG9zdC8yNjAwMTMv](https://habr.com/ru/post/478888/)). Использование этой библиотеки можно увидеть практически во всех пакетах, имеющих тестирование в том или ином виде.

```
МояПеременная = ВычислитьЧтоТоТам();  
  
// xUnit style  
Утверждения.ПроверитьРавенство(1, МояПеременная, "Моя переменная должна быть строго равна 1");  
  
// BDD style  
Ожидаем.Что(МояПеременная).Равно(1);
```

cmdline - [https://github.com/oscript-library/cmdline \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9jbWRsaW5l\)](https://github.com/oscript-library/cmdline (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9jbWRsaW5l))

Библиотека для разбора и декларативного задания аргументов командной строки.

OneScript при запуске скрипта позволяет получить все переданные скрипту аргументы через коллекцию "АргументыКоманднойСтроки", однако там они лежат в виде обычного массива строк. При разработке более сложных приложений появляется необходимость добавлять различные параметры-флаги, именованные и позиционные параметры, управлять составом команд... Тут на помощь приходит cmdline. С помощью класса ПарсерАргументовКоманднойСтроки можно указать все имеющиеся у скрипта параметры и/или команды и избавить разработчика от ненужной возни.

```
Парсер = Новый ПарсерАргументовКоманднойСтроки();  
  
Парсер.ДобавитьПараметр("ПутьКФайлу");  
Парсер.ДобавитьИменованныйПараметр("-action");  
  
Параметры = Парсер.Разобрать(АргументыКоманднойСтроки);  
Сообщить(Параметры["ПутьКФайлу"]);  
Сообщить(Параметры["-action"]);
```

В данном случае при запуске скрипта через "oscript my_script.os -action МоеДействие C:\temp\tempFile.txt" в соответствии "Параметры" окажется два значения с заранее заданными ключами - "ПутьКФайлу" будет хранить в себе "C:\temp\tempFile.txt", а "-action" - "МоеДействие".

fs -

Помощник по работе с файлами и каталогами.

Данная небольшая библиотека упрощает некоторые операции с файловой системой. В одну строку/вызов функции проверить, что файл или каталог существует, создать новый каталог, если его нет, рекурсивно скопировать все файлы из одного каталога в другой и тому подобное.

```

ФС.ОбеспечитьПустойКаталог("./build");
Если  ФС.КаталогСуществует("C:\МойРабочийКаталог") Тогда
    ФС.КопироватьСодержимоеКаталога("C:\МойРабочийКаталог", "./build");
КонецЕсли;

```

gitrunner - [https://github.com/nixel2007/gitrunner \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9naXRydW5uZXI=\)](https://github.com/nixel2007/gitrunner (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9naXRydW5uZXI=))

Библиотека-обертка над командами git из OneScript.

Предоставляет "человекочитаемое" API для управления git-репозиторием в объектной технике. Стиль работы с библиотекой похож на v8runner (о котором чуть ниже).

```

ГитРепозиторий = Новый ГитРепозиторий();
ГитРепозиторий.УстановитьРабочийКаталог(ТекущийКаталог());

ГитРепозиторий.УстановитьНастройку("core.quotePath", "true", РежимУстановкиНастроекGit.Локально);

ГитРепозиторий.ДобавитьВнешнийРепозиторий("origin", "http://github.com/EvilBeaver/oscript-library");

ГитРепозиторий.ПерейтиВВетку("master");
ГитРепозиторий.Получить();

ГитРепозиторий.ОбновитьПодмодули(Истина, Истина);

```

json - [https://github.com/oscript-library/json \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9qc29u\)](https://github.com/oscript-library/json (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9qc29u))

Библиотека работы с JSON - порт обработки Александра Переверзева.

Хотя в движке OneScript есть возможность потокового чтения JSON-строк, на мелких файлах намного удобнее (де-)сериализовать объект одной командой. Относительно авторской обработки были внесены некоторые исправления и улучшения, однако общий смысл остался тот же.

```

ПарсерJSON = Новый ПарсерJSON();

СтрокаJSON = "{
|   \"firstName\": \"Иван\",
|   \"lastName\": \"Иванов\",
|   \"address\": {
|       \"streetAddress\": \"Московское ш., 101, кв.101\",
|       \"city\": \"Ленинград\",
|       \"postalCode\": 101101
|   },
|   \"phoneNumbers\": [
|       \"812 123-1234\",
|       \"916 123-4567\"
|   ]
|}";

Объект = ПарсерJSON.ПрочитатьJSON(СтрокаJSON);

Утверждения.ПроверитьРавенство(Тип("Соответствие"), ТипЗнч(Объект));

Сообщить(Объект.firstName);

```

logos - [https://github.com/oscript-library/logos \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9sb2dvcw==\)](https://github.com/oscript-library/logos (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9sb2dvcw==))

Логирование в стиле "взрослых языков".

Логирование - большая тема любого продукта, связанного с автоматизацией деятельности. Хочется включать и отключать отладочные логи, сохранять логи в файл и выводить их в консоль, может быть даже как-то их по-хитрому форматировать, управлять настройками из файла или из переменных среды... На все эти вопросы может ответить библиотека logos. Лучше один раз показать:

```
Лог = Логирование.ПолучитьЛог("oscript.app.messages"); // Инициализация лог  
Лог.УстановитьУровень(УровниЛога.Информация); // Можно задать один из пяти уровней, например, Отладка  
  
Лог.Информация("Информационное сообщение"); // выведется в лог  
Лог.Отладка("Отладочное сообщение"); // а это - нет, т.к. уровень лога Информация выше, чем Отладка
```

strings - [https://github.com/oscript-library/strings \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9zdHJpbmdz\)](https://github.com/oscript-library/strings (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9zdHJpbmdz))

Порт модуля СтроковыеФункцииКлиентСервер на OneScript, ни больше ни меньше.

```
МассивСтрок = Новый Массив;  
МассивСтрок.Добавить("Вася");  
МассивСтрок.Добавить("пошел в");  
МассивСтрок.Добавить("Зоопарк.");  
  
// Функция должна объединить строки из массива в строку с разделителями  
РезультатСоединения = СтроковыеУтилиты.СтрокаИзМассиваПодстрок(МассивСтрок, " ");  
Ожидаем.Что(РезультатСоединения).Равно("Вася пошел в Зоопарк.");
```

tempfiles - [https://github.com/oscript-library/tempfiles \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS90ZW1wZmlsZXM=\)](https://github.com/oscript-library/tempfiles (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS90ZW1wZmlsZXM=))

Библиотека для создания временных файлов и каталогов.

Отличительной особенностью является создание файлов в некоторых "пулах", которые можно разом удалить методом ВременныеФайлы.Удалить(). Имеет два режима работы:

1. Модуль ВременныеФайлы предоставляет общий пул временных файлов среди всех используемых скриптов
2. Инициализация класса МенеджерВременныхФайлов создаст отдельный пул файлов, которым можно управлять независимо от всех остальных пулов.

```
ВременныйФайл = ВременныеФайлы.СоздатьФайл();  
ВторойВременныйФайл = ВременныеФайлы.СоздатьФайл();  
  
// мой очень полезный код  
  
ВременныеФайлы.Удалить(); // удалит оба созданных временных файла
```

tool1cd - [https://github.com/oscript-library/tool1cd \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS90b29sMWNk\)](https://github.com/oscript-library/tool1cd (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS90b29sMWNk))

Обертка над популярной утилитой для работы с файловыми базами от покинувшего нас Валерия Агеева.

Имеет два класса - ЧтениеТаблицФайловойБазыДанных и ЧтениеХранилищаКонфигурации, позволяющие вычитывать данные из файловых баз в виде таблиц значений.

```
ЧтениеБазыДанных = Новый ЧтениеТаблицФайловойБазыДанных;  
ЧтениеБазыДанных.ОткрытьФайл(ФайлХранилища);  
  
ТаблицаБД = ЧтениеБазыДанных.ПрочитатьТаблицу("VERSIONS");  
Сообщить(ТаблицаБД.Количество());  
  
ЧтениеБазыДанных.ЗакрытьФайл();
```

v8runner - [https://github.com/oscript-library/v8runner \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS92OHJ1bm5lcg==\)](https://github.com/oscript-library/v8runner (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS92OHJ1bm5lcg==))

Один из самых популярных пакетов - запуск 1С в режимах Конфигуратора и Предприятие в объектной технике.

Позволяет забыть как страшный сон десятки различных флагов и параметров для запуска 1С из командной строки и поиск нужной версии установленной платформы, инкапсулируя это внутри класса УправлениеКонфигуратором. Список доступных методов в README безбожно устарел, для осознания всей мощи инструмента стоит заглянуть в исходники.

```

УправлениеКонфигуратором = Новый УправлениеКонфигуратором();

// контекст может быть и не задан, тогда работа выполняется в базе, созданной в каталоге временных файлов
УправлениеКонфигуратором.УстановитьКонтекст("/FC:\1cdb\mydatabase","Admin", "password");

УправлениеКонфигуратором.ЗагрузитьКонфигурациюИзФайла("C:\source.cf");
УправлениеКонфигуратором.ОбновитьКонфигурациюБазыДанных();

УправлениеКонфигуратором.ВыполнитьСинтаксическийКонтроль();

```

Прочие библиотеки

Помимо библиотек из "стандартного набора" в хабе есть и другие не менее замечательные разработки.

collections - [https://github.com/ret-Phoenix/OneScript-Collections \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2JldC1QaG9lbml4L09uZVNjcmlwdC1Db2xsZWN0aW9ucw==\)](https://github.com/ret-Phoenix/OneScript-Collections (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2JldC1QaG9lbml4L09uZVNjcmlwdC1Db2xsZWN0aW9ucw==))

Расширенные коллекции для OneScript.

OneScript позволяет разрабатывать библиотеки не только на языке 1C, но и на C#. Особо желающие могут попробовать даже F#. Специально для тех, кому не хватает стандартных коллекций 1C, была создана данная библиотека. Она добавляет возможность использовать такие объекты как стек и очередь (Дек).

```

Очередь = Новый Очередь();

Очередь.Добавить("1");
Очередь.Добавить("2");
Очередь.Добавить("3");

Очередь.УбратьИзОчереди(); // в очереди останется 2 и 3

```

confluence - [https://github.com/bia-tech/os-confluence \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2JpYS10ZWNoL29zLWNvbWZsdWVuY2U=\)](https://github.com/bia-tech/os-confluence (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2JpYS10ZWNoL29zLWNvbWZsdWVuY2U=))

Взаимодействие с confluence.

Первая библиотека из категории 1C+ЧтоТо. Позволяет из OneScript управлять страницами в Confluence (собственный взгляд компании Atlassian на Wiki). Создание страниц, их редактирование, получение информации - все это через RestAPI.

```

Подключение = Confluence.ОписаниеПодключения("https://confluence.myserver.ru", "user", "password");

ИдентификаторНовойСтраницы = Confluence.СоздатьСтраницу(Подключение, "SpaceKey", "Новая страница", "Содержимое новой страниц
ы");
Confluence.ПрикрепитьМеткуКСтранице(ПараметрыПодключения, ИдентификаторНовойСтраницы, "Моя метка");

```

delegate - [https://github.com/artbear/delegate \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJlYXIVZGVsZWdhdGU=\)](https://github.com/artbear/delegate (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJlYXIVZGVsZWdhdGU=))

Делегаты/функторы в 1C.

Тем, кто скучает по Функциям как Объектам Первого Класа или кому просто хочется немного функционального программирования, может понравится эта библиотека. С ее помощью можно создать что-то вроде указателя на процедуру или функцию, передать его как параметр в какой-нибудь другой метод и там его исполнить. Является переработкой библиотеки notify.

```
// Метод, вызываемый из делегата
Процедура Поздороваться(Имя) Экспорт
    Сообщить("Привет, " + Имя + "!");
КонецПроцедуры

Делегат = Делегаты.Создать(ЭтотОбъект, "Поздороваться", "Мир");

Делегаты.Исполнить(Делегат); // или Делегат.Исполнить();

ДелегатНовыйМир = Делегаты.Создать(ЭтотОбъект, "Поздороваться");

ДелегатНовыйМир.Исполнить("Новый мир"); // или Делегаты.Исполнить(ДелегатНовыйМир, "Новый мир");
```

extensions - [https://github.com/ret-Phoenix/onescript-extensions \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L29uZXNjcmlwdC1leHRlbnNpb25z\)](https://github.com/ret-Phoenix/onescript-extensions (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L29uZXNjcmlwdC1leHRlbnNpb25z))

Дополнительные возможности для OneScript, не вошедшие в движок.

Хотя авторы очень рады любым контрибьюторам движка, некоторые возможности все же не попадают в основной движок OneScript. Данный проект - площадка для экспериментов. Возможно какие-то вещи когда-нибудь все же окажутся в основной ветке, а пока вы можете подключить данную библиотеку. Например, для работы с буфером обмена или для получения параметров экрана.

```
Буфер = Новый БуферОбмена();
Буфер.Установить("есть!");
Ожидаем,Что(Буфер.Получить()).Равно("есть!");
```

fluent - [https://github.com/nixel2007/oscript-fluent \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9vc2NyaXB0LWZsdWVudA==\)](https://github.com/nixel2007/oscript-fluent (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9vc2NyaXB0LWZsdWVudA==))

Обработка коллекций в "текущем" стиле.

Многие из авторов библиотек помимо 1С пишут на других языках. Автор данной библиотеки был "укушен Джавой", что натолкнуло его на идею обработки коллекций в "поточном" или "конвейерном" стиле.

```
Массив = Новый Массив;
Массив.Добавить(3);
Массив.Добавить(4);
Массив.Добавить(7);
Массив.Добавить(5);
Массив.Добавить(7);
Массив.Добавить(0);

Результат = ПроцессорыКоллекций.ИзКоллекции(Массив)
    .Различные()
    .Фильтровать("Результат = Элемент >= 5")
    .Обработать("Сообщить(Элемент)")
    .Обработать("Результат = Элемент + 1")
    .ВМассив();

// На выходе два сообщения в лог (5 и 7) и массив [6, 8]
```

ftp - [https://github.com/dmpas/oscript-ftp \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2RtcGFzL29zY3JpcHQiZnRw\)](https://github.com/dmpas/oscript-ftp (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2RtcGFzL29zY3JpcHQiZnRw))

FTP-клиент для OneScript.

Имя библиотеки говорит само за себя - работа с файлами по протоколу ftp.

```

Соединение = Новый FtpСоединение("example.com");

Соединение.Записать("/local/file.txt", "/dir/");

Файлы = Соединение.НайтиФайлы("/dir/", "file.txt");

Файл = Файлы[0];

РазмерФайла = Файл.Размер();

```

gui - [https://github.com/ret-Phoenix/oscript-simple-gui \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L29zY3JpcHQtc2ltcGxlLWd1aQ==\)](https://github.com/ret-Phoenix/oscript-simple-gui (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L29zY3JpcHQtc2ltcGxlLWd1aQ==))

Графический интерфейс в OneScript.

Одна из самых удивительных и неожиданных библиотек для OneScript. Добавляет возможность кодом создавать графический интерфейс для скриптов. Есть работа с таблицами и деревьями, кнопки, поля ввода и многое другое. Статья-анонс на Инфостарте от автора Сергея Ушакова - <https://infostart.ru/public/540284/> (<https://infostart.ru/public/540284/>)

Отдельно добавлю, что при подключении установленной из хаба библиотеки, подключать внешнюю компоненту руками уже не нужно.

```

#Использовать gui

Перем Форма;

//# Обработка события первого открытия формы
Процедура ПриОткрытииФормы() Экспорт

    ПолеВвода1 = Форма.Элементы.Добавить("ПолеВвода1", "ПолеФормы", Неопределено);
    ПолеВвода1.Вид = Форма.ВидПоляФормы.ПолеВвода;
    ПолеВвода1.Заголовок = "Поле ввода";
    ПолеВвода1.Значение = "Значение задано из скрипта";

КонецПроцедуры

УправляемыйИнтерфейс = Новый УправляемыйИнтерфейс();
Форма = УправляемыйИнтерфейс.СоздатьФорму();

//# Устанавливаем обработку события ПриОткрытии
Форма.УстановитьДействие(ЭтотОбъект, "ПриОткрытии", "ПриОткрытииФормы");
Форма.Показать();

```

InternetMail - [https://github.com/dmpas/oscript-mail \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2RtcGFzL29zY3JpcHQtbWFpbA==\)](https://github.com/dmpas/oscript-mail (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2RtcGFzL29zY3JpcHQtbWFpbA==))

Реализация протокола SMTP для отправки почты.

Повторение синтаксиса и классов 1С по работе с почтовыми сообщениями, но на C# и из OneScript.

```

Сообщение = Новый ИнтернетПочтовоеСообщение;
Сообщение.Получатели.Добавить("receiver@mail.ru");
Сообщение.ОбратныйАдрес.Добавить("someuser@mail.ru");
Сообщение.Отправитель = "someuser@mail.ru";
Сообщение.Тема = "Server is down";
Текст = "
    |<h3>Server is down !</h3>
    |Надо что-то делать.<br/>";

Сообщение.Тексты.Добавить(Текст, ТипТекстаПочтовогоСообщения.HTML);
Сообщение.Вложения.Добавить("C:/Пример вложения 1.docx");
Сообщение.Вложения.Добавить(
    Новый ДвоичныеДанные("C:/Пример вложения 2.docx"),
    "Пример вложения 2.docx"
);

```

messenger - [https://github.com/oscript-library/messenger \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9tZXNzZW5nZXI=\)](https://github.com/oscript-library/messenger (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9tZXNzZW5nZXI=))

Отправка сообщений в различные сервисы.

Еще одна интеграционная библиотека. На этот раз - для отправки сообщений. На момент написания статьи умеет отправлять SMS (через трех операторов), а так же писать сообщения в Slack, RocketChat, Gitter и Telegram. Слышал о "success story" с отправлением ошибок из Журнала Регистрации в Телеграм-чат с последующей раздачей по шапке получившим сообщение.

```
ИмяКомнаты = "organization/repo";
Мессенджер = Новый Мессенджер();
Мессенджер.ИнициализацияGitter(ТокенПользователя);
Мессенджер.ОтправитьСообщение(Мессенджер.ДоступныеПротоколы().gitter, ИмяКомнаты, "Всем привет!" );
```

notify - [https://github.com/nixel2007/notify \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9ub3RpZnk=\)](https://github.com/nixel2007/notify (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9ub3RpZnk=))

Создание и обработка объектов ОписаниеОповещения.

Не смотря на то, что асинхронных функций как таковых в OneScript нет, объект ОписаниеОповещения и идущий с ним в паре метод ВыполнитьОбработкуОповещения довольно хорошо подошли под реализацию Функций как Объектов Первого Класа, сохранив максимально приближенный к 1С синтаксис. Можно использовать, например, для реализации паттерна "функции обратного вызова" (коллбэка).

```
// Процедура-обработчик описания оповещения.
//
Процедура СообщитьПриветМир(Результат, ДополнительныеПараметры = Неопределено) Экспорт
    Сообщить("Привет, " + ДополнительныеПараметры + "!");
    ВызватьИсключение "Что-то произошло!";
КонецПроцедуры

// Создание объекта ОписаниеОповещения. Аналогично использованию "Новый ОписаниеОповещения("СообщитьПриветМир", ЭтотОбъект, "Мир")" в 1С.
ОписаниеОповещения = ОписанияОповещений.Создать("СообщитьПриветМир", ЭтотОбъект, "Мир");

// Выполнение обработки оповещения. Аналогично использованию ВыполнитьОбработкуОповещения(ОписаниеОповещения) в 1С.
ОписанияОповещений.ВыполнитьОбработкуОповещения(ОписаниеОповещения);
```

params - [https://github.com/artbear/params \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJlYXVvcGFyYW1z\)](https://github.com/artbear/params (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJlYXVvcGFyYW1z))

Стандартизированное переопределение параметров командной строки из json-файлов конфигурации и переменных окружения.

Данная библиотека условно является надстройкой над библиотекой "cmdline". Используя вывод ПарсераАргументовКоманднойСтроки, анализируя наличие и содержимое файла настроек и переменных среды, позволяет получить итоговые значения параметров. У каждого типа задания параметра есть свой приоритет. Файлы настроек могут быть вложенными друг в друга через специальные поля-ссылки. Является продолжением идей библиотеки ReadParams.

```
Парсер = Новый ПарсерАргументовКоманднойСтроки();
Парсер.ДобавитьИменованныйПараметр("Параметр");

Параметры = ЧтениеПараметров.Прочитать(Парсер, АргументыКоманднойСтроки);
```

ParserFileV8i - [https://github.com/ret-Phoenix/parserV8i \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L3BhcnNlclY4aQ==\)](https://github.com/ret-Phoenix/parserV8i (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L3BhcnNlclY4aQ==))

Чтение и запись файла списка баз **.v8i**.

Обработка списка баз в виде соответствия, поиск баз по имени или пути. На сладкое - класс для очистки кэша.

```
Парсер = Новый ПарсерСпискаБаз;
Парсер.УстановитьФайл("tests/ibases.v8i");

Результат = Парсер.НайтиПоПути("http://localhost/lessons/");

Ожидаем.Что(Результат.Connect.String).Павно("Connect=ws=""http://localhost/lessons/""");
```


progbars - [https://github.com/theshadowco/progbars \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3RoZXNoYWRvd2NvL3Byb2diYXI=\)](https://github.com/theshadowco/progbars (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3RoZXNoYWRvd2NvL3Byb2diYXI=))

Консольный прогресс-бар.

В OneScript по сравнению с 1С добавлены возможности по взаимодействию с окном консоли. Перемещение курсора, перерисовка и прочие милые сердцу старых паскалистов/сишников вещи. Данная библиотека дает удобную возможность по рисованию и управлению консольным прогресс-баром.

```
ПрогрессБар = Новый ПрогрессБар();
ПрогрессБар.Начать(100, "Прогресс ");

Для Ит = 0 По 100 Цикл

    ПрогрессБар.СделатьШаг();
    Приостановить(200);

КонецЦикла;
ПрогрессБар.Завершить();
```

ReadParams - [https://github.com/Stepa86/ReadParams \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL1N0ZXBhODYvUmVhZFBhcmFtcw==\)](https://github.com/Stepa86/ReadParams (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL1N0ZXBhODYvUmVhZFBhcmFtcw==))

Чтение параметров скрипта из json-файла.

Выполняет десериализацию JSON-файла настроек в соответствие, поддерживает включение параметров из одного файла в другой через специальные поля-ссылки, а так же подстановку значений одних полей в другие.

```
записьТекста = Новый ЗаписьТекста(файлПараметров1);
ЗаписьТекста.ЗаписатьСтроку( "{""парам.Число"": 1, ""парам.Строка"": ""1"", ""парам.Булево"": true}" );
ЗаписьТекста.Закрыть();

прочитанныеПараметры = ЧтениеПараметров.Прочитать( файлПараметров1 );

// прочитанныеПараметры["парам.Число"] = 3;
// прочитанныеПараметры["парам.Строка"] = "3";
// прочитанныеПараметры["парам.Булево"] = Истина;
```

restler - [https://github.com/oscript-library/restler \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9yZXN0bGVy\)](https://github.com/oscript-library/restler (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9yZXN0bGVy))

Помощник работы с RestAPI.

Довольно молодая библиотека, родившаяся после нескольких мучительных дней работы с HTTPсоединением. Упрощает шаги по вызову различных http-сервисов, сама делает десериализацию, позволяет управлять кодом состояния. Имеет бедное README, однако уже зарекомендовала себя (личным примером) как отличный фундамент для построения более высокоуровневых интеграционных компонентов уже как минимум к пяти различным системам.

```
Клиент = Новый КлиентВебAPI();
Клиент.ИспользоватьСоединение(Новый HTTPСоединение("localhost"));
ИнформацияОПродажах = Клиент.Получить("sales/totals");

Для Каждого Инфо Из ИнформацияОПродажах Цикл
    Сообщить(СтрШаблон("Продано товара %1 на сумму %2", Инфо["nomenclature"], Инфо["sum"]));
КонецЦикла;
```

sql - [https://github.com/ret-Phoenix/oscript-sql \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L29zY3JpcHQtc3Fs\)](https://github.com/ret-Phoenix/oscript-sql (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3JldC1QaG9lbml4L29zY3JpcHQtc3Fs))

Работа с SQL-базами из OneScript.

Изначально библиотека разрабатывалась для работы с sqlite - чтение и обработка Журнала Регистрации и собственные мини-базы (в том числе in-memory). На данный момент дополнительно поддерживаются MS SQL Server, MySQL и PostgreSQL.

```
Соединение = Новый Соединение();
Соединение.ТипСУБД = Соединение.ТипыСУБД.sqlite;
Соединение.ИмяБазы = ":memory:";
Соединение.Открыть();

Запрос = Новый Запрос();
Запрос.УстановитьСоединение(Соединение);
Запрос.Текст = "Create table users (id integer, name text)";
Запрос.ВыполнитьКоманду();

Запрос.Текст = "insert into users (id, name) values(1, @name)";
Запрос.УстановитьПараметр("name", "Сергей");
Запрос.ВыполнитьКоманду();

Запрос2 = Новый Запрос();
Запрос2.УстановитьСоединение(Соединение);
Запрос2.Текст = "select * from users where id = @id";
Запрос2.УстановитьПараметр("id", 1);
T3 = Запрос2.Выполнить().Выгрузить();
```

v8storage - [https://github.com/oscript-library/v8storage \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tLT29zY3JpcHQtOGlicmFyeS92OHN0b3JhZ2U=\)](https://github.com/oscript-library/v8storage (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tLT29zY3JpcHQtOGlicmFyeS92OHN0b3JhZ2U=))

Работа с Хранилищем 1С в стиле v8runner.

Библиотека является развитием идей и функциональности `v8runner`, специализируясь именно на работе с хранилищем.

```
ХранилищеКонфигурации = Новый МенеджерХранилищаКонфигурации();
ХранилищеКонфигурации.УстановитьКаталогХранилища(КаталогХранилища);

ХранилищеКонфигурации.ПрочитатьХранилище();

ХранилищеКонфигурации.СохранитьВерсиюКонфигурацииВФайл(НомерВерсии, ИмяФайлаКонфигурации);

ТаблицаВерсий = ХранилищеКонфигурации.ПолучитьТаблицаВерсий();
МассивАвторов = ХранилищеКонфигурации.ПолучитьАвторов();
```

v8unpack - [https://github.com/dmpas/oscript-v8unpack \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2RtcGFzL2Z3Y3JpcHQtdjh1bnBhY2s=\)](https://github.com/dmpas/oscript-v8unpack (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2RtcGFzL2Z3Y3JpcHQtdjh1bnBhY2s=))

Порт компоненты для работы с контейнерами файлов 1С:Предприятие 8.

Среди людей, которые хоть раз занимались распаковкой и анализом внешних обработок или конфигураций 1С (еще до 8.3.7), утилита `v8unpack (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tLTJtRGFzL3Y4dW5wYWNR)` давно известна. Данный порт является переработкой данного приложения в виде компоненты на C# для вызова ее изнутри OneScript без всяких `ЗапуститьПриложение()`. На данный момент поддерживается только команда `-parse` в тех рамках, в которых она используется в приложении `qitsync` (о нем ниже).

```
#Использовать v8unpack

ЧтениеФайла = Новый ЧтениеФайла8("ПутьКФайлу");
Для Каждого мЭлемент Из ЧтениеФайла.Элементы Цикл
    Сообщить(мЭлемент.Имя);
    ЧтениеФайла.Извлечь(мЭлемент, "Каталог", Истина);
КонецЦикла;
```

vanessa-behavior - [https://github.com/silverbulleters/vanessa-behavior2 \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3NpbHZlcmJ1bGxldGVycy92YW5lc3NhLWJlIGF2aW9yMg==\)](https://github.com/silverbulleters/vanessa-behavior2 (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3NpbHZlcmJ1bGxldGVycy92YW5lc3NhLWJlIGF2aW9yMg==))

Фреймворк для разработки через BDD, упакованный в виде пакета.

Не совсем "библиотека", но по формальному признаку (отсутствие оберток для запуска из командной строки и наличия подключаемых модулей) отнесена в этот раздел. Используется для быстрой установки Vanessa Behavior на машину разработчика или в локальный каталог репозитория через команду **opm install -i vanessa-behavior**. Обращаю внимание, что данный пакет собирается из репозитория Vanessa-Behavior2, и может отставать по релизом от (пока еще) основного репозитория.

yadisk - [https://github.com/kuntashov/oscript-yadisk \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2t1bnRhc2hvdj9vc2NyaXB0LXIhZGlzaw==\)](https://github.com/kuntashov/oscript-yadisk (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2t1bnRhc2hvdj9vc2NyaXB0LXIhZGlzaw==))

Работа с Яндекс.Диск через RestAPI.

С помощью этого пакета можно автоматизировать загрузку/выгрузку файлов с помощью Яндекс.Диск, получать свойства диска (размер, свободное и занятое пространство), работать с корзиной и прочее. Вот здесь (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2t1bnRhc2hvdj9vc2NyaXB0LXIhZGlzay11cGxvYWRLcg==) есть пример использования библиотеки от самого автора - отдельное приложение для загрузки файлов.

```
OAuth_Токен = "..."; // Используйте полученный токен для вашего приложения.
```

```
ЯндексДиск = Новый ЯндексДиск;  
ЯндексДиск.УстановитьТокенАвторизации(OAuth_Токен);
```

```
ИмяФайла = Строка(Новый УникальныйИдентификатор) + ".txt";  
ПутьНаДиске = "/test-folder/" + ИмяФайла;
```

```
ЯндексДиск.ЗагрузитьНаДиск("c:/data/file-to-upload.txt"), ПутьНаДиске);
```

Приложения

Как я писал выше, приложения - это те же самые скрипты для интерпретатора OneScript, но предназначенные для запуска из командной строки. При установке таких приложений через орм в PATH добавляются обертки в виде bat/sh-файлов, позволяющие запускать их без лишних телодвижений. Например, вместо строки

```
"C:\Program Files (x86)\OneScript\bin\oscript.exe" "C:\Program Files (x86)\OneScript\lib\gitsync\src\gitsync.os"
```

можно просто вызвать приложение "gitsync". Целью создания таких приложений является упрощение выполнения каких-то типовых задач "в одну строку", без написания собственных скриптов. Часто эти же запуск этих приложений встраивается в сборочные линии, например, на Jenkins CI. Аналогично библиотекам, часть приложений поставляется в "стандартном наборе", остальные доступны для установки через орм.

Поставляемые приложения

1bdd - [https://github.com/artbear/1bdd \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJIYXVhZGJkZA==\)](https://github.com/artbear/1bdd (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJIYXVhZGJkZA==))

Фреймворк для разработки через BDD в OneScript.

Приложение позволяет запускать проверки поведения по заранее написанным feature-файлам. Реализации шагов (step_definitions) описываются в виде скриптов для OneScript. Имеются возможности по пакетному запуску прогона фич, генерации реализаций шагов по feature-файлам, представление результатов отчета в формате junit, собственная библиотека шагов. Пакет может быть подключен как библиотека через #Использовать, например, для реализации программного запуска прогона сценариев и файлов задач (tasks) пакетного менеджера орм.

```
1bdd ./features -junit-out out/
```

1testrunner - [https://github.com/artbear/1testrunner \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJIYXVhZGJkZA==\)](https://github.com/artbear/1testrunner (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2FydGJIYXVhZGJkZA==))

Фреймворк для разработки через TDD в OneScript.

Пакет является портом фреймворка xUnitFor1C. Предназначен для запуска tdd-тестов. Реализация тестов описывается в виде скриптов для OneScript. Можно запускать отдельные тесты или каталог с тестами целиком. Так же, как и в 1bdd, есть представление результатов тестирования в виде отчета junit и подключение как библиотеки через #Использовать.

```
# запустит все тесты из каталога tests и сформирует отчет о результатах тестирования в каталоге build  
1testrunner -runall ./tests xddReportPath ./build
```

gitsync - [https://github.com/oscript-library/gitsync \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9naXRzeW5j\)](https://github.com/oscript-library/gitsync (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9naXRzeW5j))

Синхронизация хранилища 1C с репозиторием git.

Наверное, самое известное приложение для OneScript. Осуществляет выгрузку каждого помещения в хранилище в виде коммита в git-репозиторий, переносит автора помещения, дату и текст сообщения. После начальной инициализации репозитория служебными файлами (указатель на версию хранилища и данные об авторах) выгрузку можно запустить командой

```
gitsync export c:/storage/zup src/cf
```

Прочие приложения

cpdb - [https://github.com/arkuznetsov/cpdb \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2Fya3V6bmV0c292L2NwZGI=\)](https://github.com/arkuznetsov/cpdb (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2Fya3V6bmV0c292L2NwZGI=))

Приложение-комбайн по работе базой данных в файловом формате и MS SQL Server.

Позволяет автоматизировать загрузку/выгрузку базы, отправку базы на Yandex.Диск, работу с сетевыми дисками, выполнять архивацию и прочие частые операции. С полным списком команд можно ознакомиться в README проекта.

Пример выполнения бэкапа и отправки его на Я.Диск:

```
cpdb backup -sql-srvr MySQLName MyDatabase -sql-user sa -sql-pwd 12345 -bak-path "d:\MSSQL\Backup\MyDatabase_copy.bak"
cpdb putyadisk -file "d:\MSSQL\Backup\MyDatabase_copy.bak" -ya-token XXXXXXXXXXXXXXXXXXXXXXXXXX -ya-path "/transfer/MyDatabase_copy.bak" -delsrc
```

deployka - [https://github.com/oscript-library/deployka \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9kZXBs3lrYQ==\)](https://github.com/oscript-library/deployka (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9kZXBs3lrYQ==))

Развертывание конфигураций 1C.

Приложение позволяет автоматизировать обновление конфигураций из хранилища/файла конфигурации/файла обновления. Приятным дополнением является возможность управления активными сеансами через родные для 1C утилиты ras/ras.

```
# Блокировка базы с удалением всех сеансов
deployka session kill -ras myserver:1545 -db-user admin -db-pwd pass -lockucode "123"
# Обновление конфигурации из хранилища
deployka loadrepo /FC:\Infobase C:\Хранилище1C -db-user admin -db-pwd pass -storage-user admin -storage-pwd pass -ucode "123"
# Обновление БД
deployka dbupdate /FC:\Infobase -db-user admin -db-pwd pass -ucode "123"
# Разблокировка базы
deployka session unlock -ras myserver:1545 -db-user admin -db-pwd pass
```

oscript-config - [https://github.com/nixel2007/oscript-config \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9vc2NyaXB0LWNvbmZpZW==\)](https://github.com/nixel2007/oscript-config (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL25peGVsMjAwNy9vc2NyaXB0LWNvbmZpZW==))

Работа с файлом конфигурации OneScript oscript.cfg

Позволяет зачитывать и устанавливать настройки приложения oscript.exe из командной строки. Этот пакет необходим для работы системы автодополнения кода по библиотекам OneScript в редакторе кода Visual Studio Code.

```
> oscript-config get lib.system
C:\Program Files (x86)\OneScript\lib

> oscript-config set lib.additional "./oscript_modules"
```

packman - [https://github.com/oscript-library/packman \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9wYWNRbWFWu\)](https://github.com/oscript-library/packman (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL29zY3JpcHQtbGlicmFyeS9wYWNRbWFWu))

Сборка тиражных релизов конфигураций 1C:Предприятие 8.

Данное приложение позволяет автоматизировать создание поставки конфигурации, как в виде файлов поставки, так и в виде setup.exe. Дополнительно получившийся дистрибутив можно упаковать в zip-архив.

Простой пример работы:

```
packman load-storage C:/Хранилище1C -storage-user admin -storage-pwd pass
packman make-cf
packman МанифестПоставки.edf -out ./build -setup
packman zip-dist -in ./build -out ./dist
```

precommit1c - [https://github.com/xDrivenDevelopment/precommit1c \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3hEcml2ZW5EZXXZlG9wbWVudC9wcmVjb21taXQxYw==\)](https://github.com/xDrivenDevelopment/precommit1c (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3hEcml2ZW5EZXXZlG9wbWVudC9wcmVjb21taXQxYw==))

Приложение для разборки внешних обработок при выполнении коммита в git.

Помимо разборки обработок, данный пакет умеет собирать обработки обратно. Отличительной особенностью использования утилиты в виде пакета является возможность установки всех необходимых файлов в репозиторий в одну команду:

```
precommit1c --install
```

scenex - [https://github.com/arkuznetsov/scenex \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2Fya3V6bmV0c292L3NjZW5leA==\)](https://github.com/arkuznetsov/scenex (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL2Fya3V6bmV0c292L3NjZW5leA==))

Исполнитель сценариев, объявленных декларативно в JSON-файле.

Позволяет сформировать один небольшой конфигурационный файл, а затем махом выполнять все описанные в нем сценарии.

Запуск:

```
scenex batch "./deploy_scenario.json"
```

Пример файла сценариев:

```
{
  "stages": {
    "Вывод справки oscript": {
      "description": "Вывод справки oscript",
      "tool": "oscript",
      "command": ""
    },
    "Вывод справки deployка (команда loadcfg)": {
      "description": "Вывод справки deployка",
      "tool": "deployка",
      "command": "help",
      "params": [
        "loadcfg"
      ]
    }
  }
}
```

vanessa-runner - [https://github.com/silverbulleters/vanessa-runner \(/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3NpbHJlcmJ1bGxldGVycy92YW5lc3NhLXJ1bm5lcg==\)](https://github.com/silverbulleters/vanessa-runner (/redirect.php?url=aHR0cHM6Ly9naXRodWIuY29tL3NpbHJlcmJ1bGxldGVycy92YW5lc3NhLXJ1bm5lcg==))

Автоматизация различных операций 1C-разработчика и dev-ops.

Приложение является большим комбайном из возможностей, частично портированных из других библиотек. Умеет разбирать и собирать файлы cf/cfe/epf/erf, упрощает запуск тестирования с помощью фреймворков Vanessa Behavior и xUnitFor1C, создавать и обновлять информационные базы, выполнять миграцию данных конфигураций, построенных на БСП, запускать синтаксический контроль с выводом результата проверки в формат junit, делать привязку баз к хранилищу. Полный список возможностей лучше изучать через вывод команды **vranner help**. Конфигурируется через параметры командной строки, переменные среды и файл настроек в формате JSON.

```
# Выполнение миграции
vrunner run --command "ЗапуститьОбновлениеИнформационнойБазы;ЗавершитьРаботуСистемы;" --execute $runnerRoot/epf/ЗакрытьПредприя
тие.epf --ibname /FC:\ИнфоБаза1C
# Запуск проверки поведения с помощью фреймворка Vanessa Behavior
vrunner vanessa --settings tools/vrunner.json --ibname /FC:\ИнфоБаза1C
```

Заключение

Данным обзором я постарался разом ответить на большинство вопросов вида "А есть ли библиотека/приложение для вот этого". Если вдруг готовой библиотеки не оказалось, не бойтесь писать свои и выкладывайте их на GitHub. Сообщество будет вам крайне благодарно :) Обсуждение имеющихся библиотек сосредоточено в Gitter-канале EvilBeaver/oscript-library (/redirect.php?url=aHR0cHM6Ly9naXR0ZXIuaW0vRXZpbEJlYXZlci9vc2NyaXB0LWxpYnJhcnk=) и в Issues каждой конкретной библиотеки.

В следующей статье я опишу как создавать собственные библиотеки и приложения, готовые для публикации в хабе пакетов OneScript.

Спасибо за внимание!

P.S. В последнем релизе расширения поддержки языка 1C в VSCode появился полноценный Синтакс-Помощник по всем классам и модулям всех установленных библиотек. При наличии README в пакете выводится и он. Обновляйтесь, нажимайте ctrl-f1 и выбирайте в выпадающем меню новый раздел oscript-library. Так же не забывайте ставить пакет oscript-config для получения частичной поддержки автодополнения по модулям и классам библиотек прямо при наборе кода.

Полная версия (/public/699642/)

© ООО "Инфостарт", 2006-2013 [www.infostart.ru](http://infostart.ru) (<http://infostart.ru>)

