General Conventions

# Формат сообщений commit

Сообщения должны быть на английском ( North American English) языке.

**<type>(<scope>): <subject>**

<Одна пустая строка>

**<body>**

<**type**> - это одно из следующих слов:

* feat (feature)
* fix (bug fix)
* docs (documentation)
* style (formatting, missing semi colons, …)
* refactor
* test (when adding missing tests)
* chore (maintain)

<**scope**> - это логическая область commit, например, имя модуля, docs и т.п.

<**subject**> - должен начинаться с глагола в повелительном наклонении со строчной ( маленькой) буквы и не должен занимать более 80 символов – при написании стоит кратко описать суть доработки. В детали имеет смысл вдаваться в body. В то же время описание не должно быть абстрактным: например, fix an issue, optimize performance. Здесь стоит уточнить: fix the issue with wrong payment destination, optimize performance by reducing interval of system check. <subject> не должен заканчиваться многоточием.

<**body**> - необязательная часть. Нужна в случае если требуются детали. Должна отвечать на вопросы “Что? Почему? Зачем? Как?”.

**Примеры сообщений** ( можно также посмотреть в проекте с помощью git log).

**feat(OraMakeSystem): add module OraMakeSystem - automation of Oracle modules development**

**fix($compile): couple of unit tests for IE9**

Older IEs serialize html uppercased, but IE9 does not...

Would be better to expect case insensitive, unfortunately Jasmine does

not allow to user regexps for throw expectations.

# Теги в git

В качестве аналога тегов модулей в SVN в git могут служить аннотированные теги (annotated tags)[[1]](#footnote-1). Тег создаётся для обозначения некоторой вехи в развитии модуля – как правило, стабильной версии, рекомендуемой для использования в прикладной разработке. По сравнению с SVN, теги в git являются легкими: их создание не требует операций копирования файлов; под их создание практически не расходуется место на жёстком диске локальной машины или в репозитории. Фактически, тег в git полностью соответствует своему названию (tag) – это метка, которой помечается коммит. При создании тега, помимо его имени, в базе данных git сохраняется следующая информация:

* наименование тега;
* контрольная сумма (SHA-1);
* автор тега;
* дата и время создания тега;
* описание тега.

**Тег создаётся для репозитория целиком**. В отличие от SVN, в git нельзя выбрать папку, на основе которой делается тег. В связи с этим правила именования тегов пересматриваются.

# Порядок создания и именования тегов

Если изменению с момента создания последнего тега изменениям подвергся лишь один модуль (например, только JepRia), то имя тега формируется следующим образом:

Module-X.Y.Z, где Module – имя модуля. Пример: JepRia-8.5.0

Если ранее был создан тег для установки определенной версии модуля и затем пришлось внести изменения в процесс установки или документацию по установке, которые не оказывают влияния на результат успешной установки ( т.е. нет причин менять версию модуля), то в имя тега через тире добавляется порядковый номер изменения процесса установки ( начиная с 1):

Module-X.Y.Z-Q, где Q – порядковый номер изменения установки. Пример: Option-3.3.0-1

Если изменению подвергались одновременно несколько модулей, то в теге их названия и номера указываются через запятую:

Module1-X1.Y1.Z1,Module2-X2.Y2.Z2,… Пример: JepRia-8.6.0,JepRiaToolkit-6.2.0

Примечание: **имя тега в git не может содержать пробела**, поэтому название модуля отделяется от версии знаком дефиса.

В описание (message) тега следует писать release notes – так же, как это для тегов в SVN, но с использованием новых соглашений по commit-сообщениям.

После создания тега из каждого модуля необходимо формировать отдельные zip-архивы, включающие в себя папки DB, App, Doc. Папку App необходимо очистить от «мусора», т.е. логов и временных файлов сборки. Имена файлов при этом формируются следующим образом: Module-X.Y.Z.zip. Так, в примере выше с двумя модулями в теге будет создано два файла: JepRia-8.6.0.zip и JepRiaToolkit-6.2.0.zip.

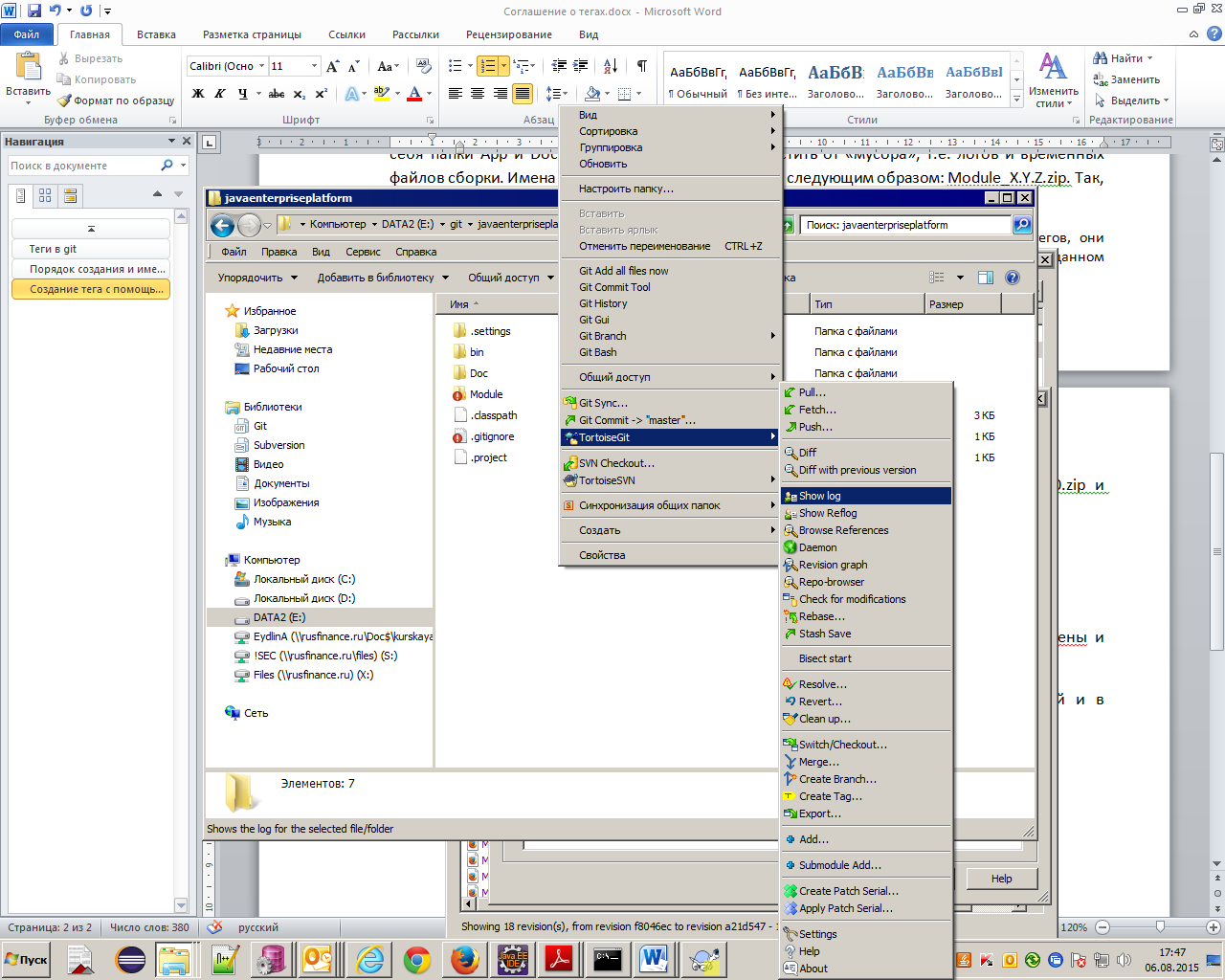
Созданные архивы размещаются в

https://github.com/oramake/oramake-framework/releases.

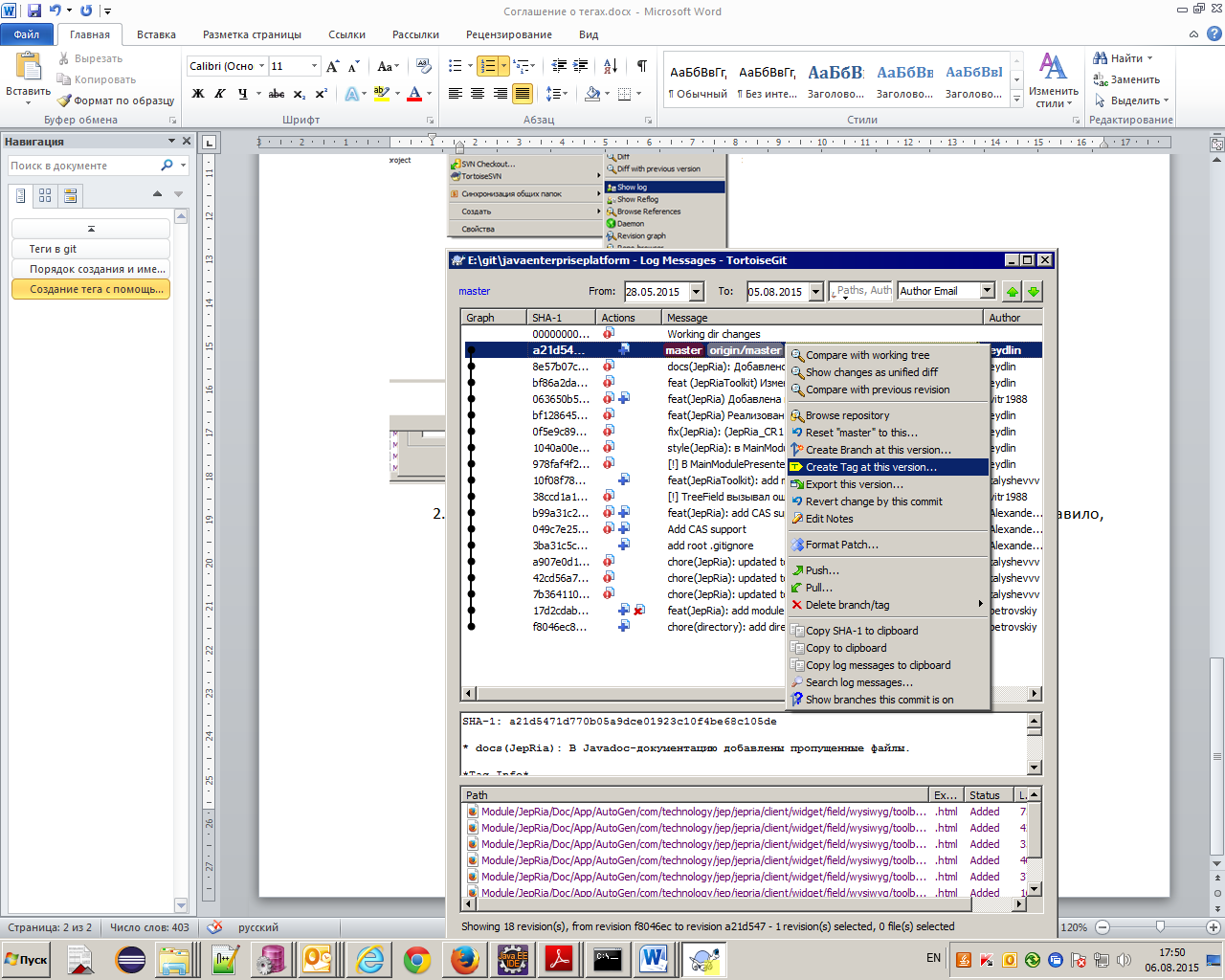
# Создание тега с помощью TortoiseGit

Предполагается, что все планируемые к включению в тег изменения закомичены и отправлены в репозиторий.

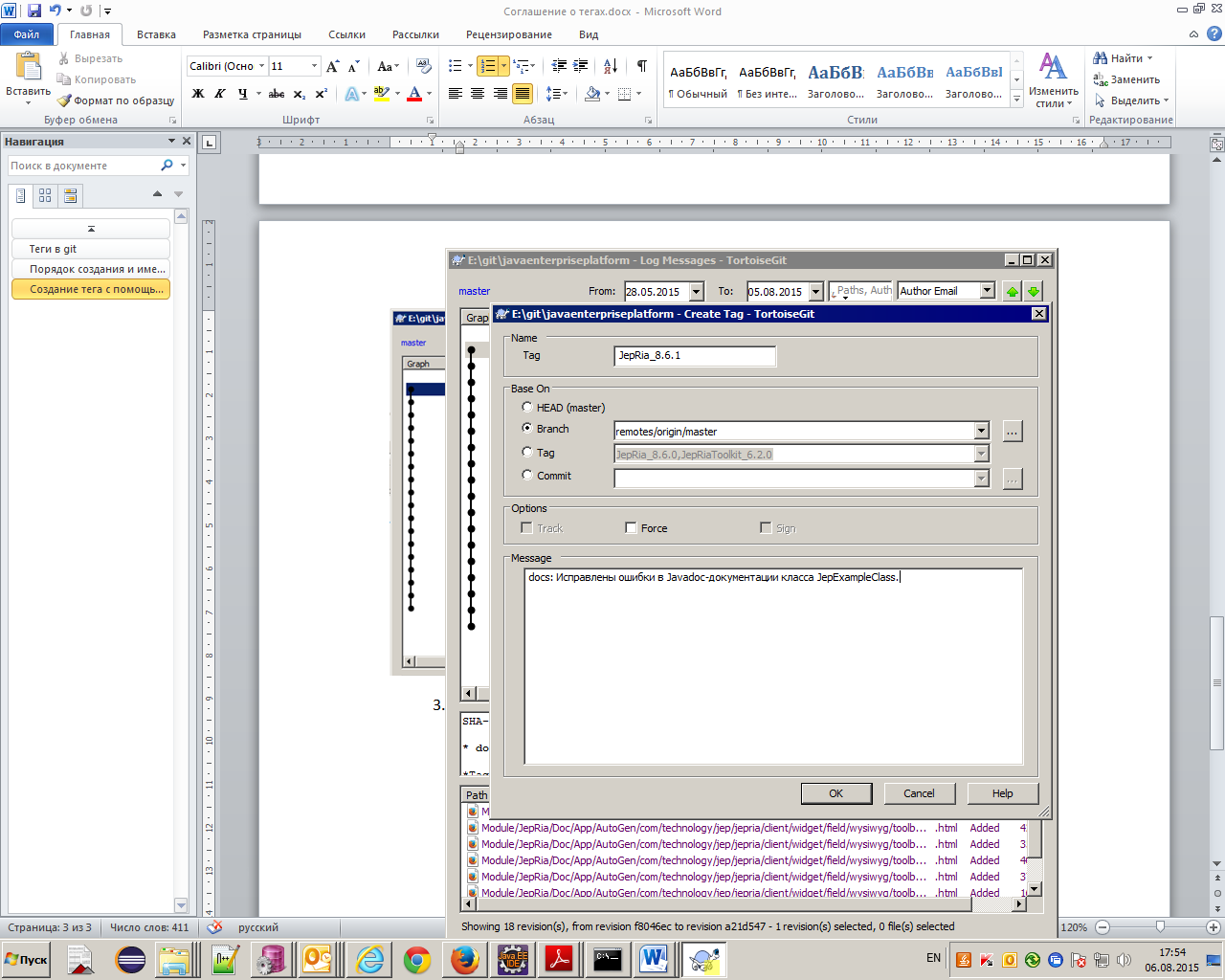
1. В папке репозитория на локальной машине необходимо кликнуть правой кнопкой и в открывшемся меню выбрать *TortoiseGit -> Show log…*



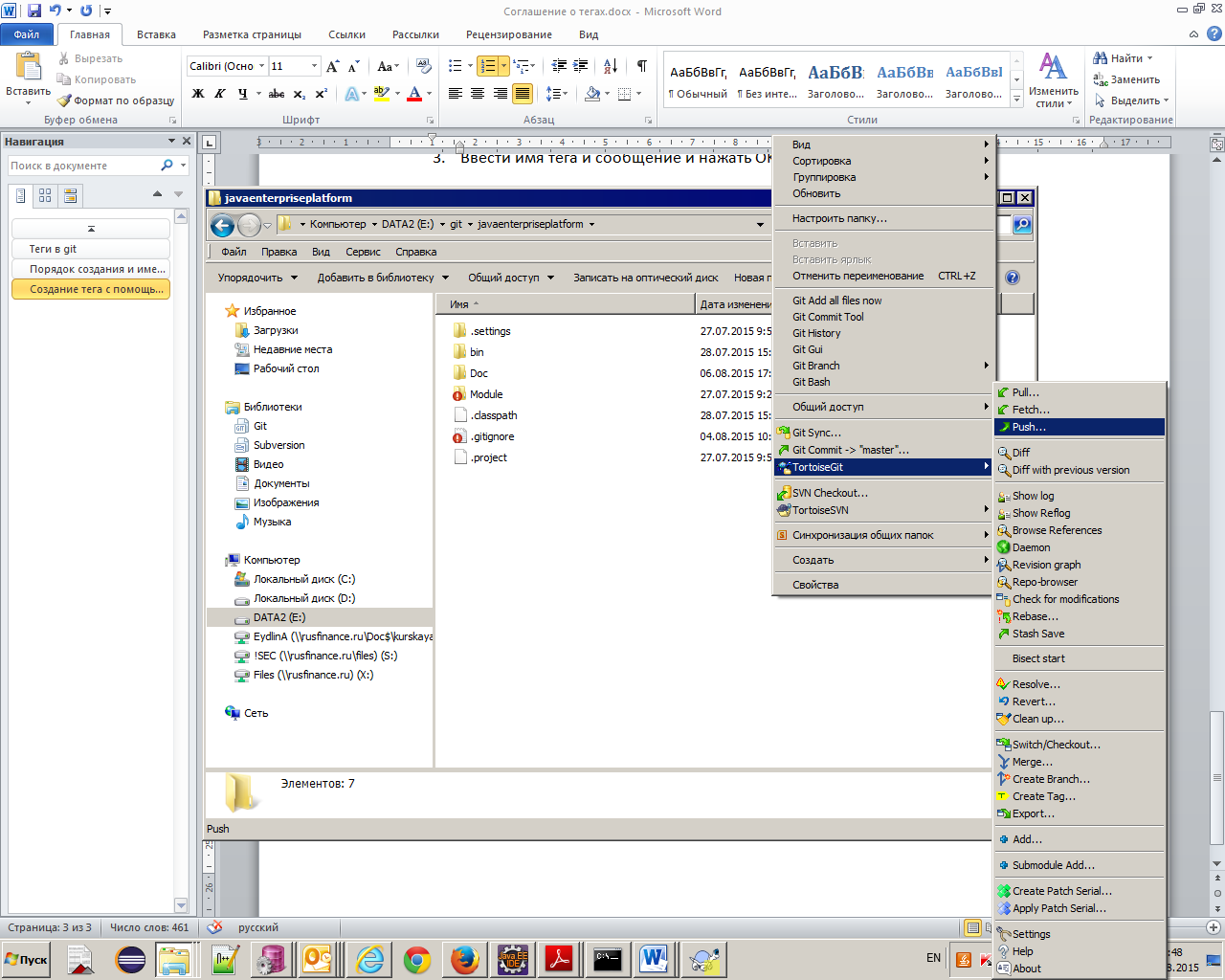
1. В открывшемся окне выбрать коммит, на основе которого создаётся тег (как правило, это будет master), кликнуть правой кнопкой и выбрать *Create Tag at this version…*



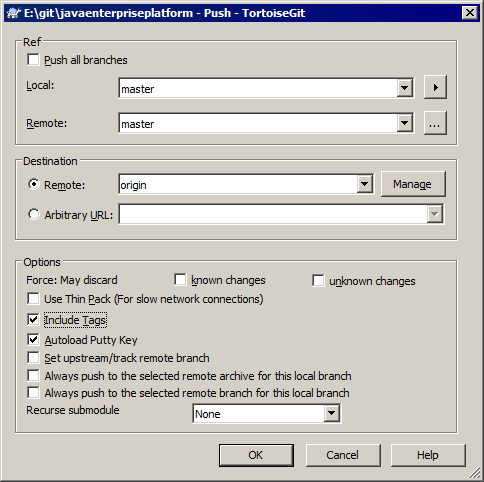
1. Ввести имя тега и сообщение и нажать *OK*.



1. Тег был создан локально, **в репозитории его пока нет**. Необходимо отправить его в репозиторий (push). Для этого необходимо в папке репозитория на локальной машине кликнуть правой кнопкой мыши и выбрать *TortoiseGit -> Push…*



1. В появившемся окне необходимо **поставить флаг** *Include tags* и нажать OK.



**Примечание:** Вместо шагов 4–5 можно воспользоваться открыть командную строку, перейти в папку репозитория и воспользоваться командой:

git push origin –tags

# Источники:

* один из примеров конвенций [https://docs.google.com/document/d/1QrDFcIiPjSLDn3EL15IJygNPiHORgU1\_OOAqWjiDU5Y/edit?pli=1#](https://docs.google.com/document/d/1QrDFcIiPjSLDn3EL15IJygNPiHORgU1_OOAqWjiDU5Y/edit?pli=1%23)
* Статья <http://chris.beams.io/posts/git-commit/>
* Небольшая статья на тему <http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html>
* Юмор про git commit massages: <http://xkcd.com/1296/>

1. Существуют также «легковесные» теги (lightweight tag). В отличие от аннотированных тегов, они представляют собой просто указатель на коммит; под них не создаётся объект в базе данных git. В данном документе они **не** рассматриваются. [↑](#footnote-ref-1)