Подготовка релизов FreeBSD

Аннотация

В этой статье описывается процесс разработки релизов проекта FreeBSD.

Содержание

1. Введение в процесс разработки релизов FreeBSD . . . 1
2. Общая информация и подготовка . . . 2
3. Терминология выпуска релизов . . . 5
4. Изменения на веб-сайте в течение цикла выпуска . . . 5
5. Выпуск релиза от main . . . 7
6. Выпуск релиза от stable/ . . . 8
7. Создание установочных носителей FreeBSD . . . 10
8. Публикация установочных носителей FreeBSD на зеркалах проекта . . . 12
9. Завершение цикла выпуска . . . 13
10. Конец срока поддержки выпуска . . . 14

1. Введение в процесс разработки релизов FreeBSD

Разработка FreeBSD следует очень специфичному рабочему процессу. В общем случае все изменения в базовой системе FreeBSD вносятся в ветку main, которая отражает вершину дерева исходного кода.

После достаточного периода тестирования изменения могут быть объединены в ветки stable/. Минимальный срок по умолчанию перед объединением в ветки stable/ составляет три (3) дня.

Хотя общее правило предписывает выждать минимум три дня перед слиянием из main, существуют особые обстоятельства, при которых может потребоваться немедленное слияние, например, критическое исправление безопасности или исправление ошибки, напрямую блокирующей процесс сборки выпуска.

Через несколько месяцев, когда количество изменений в ветке stable/ значительно увеличилось, наступает время выпуска следующей версии FreeBSD. Исторически такие выпуски называются релизами "с точкой".

Между выпусками из ветвей stable/, примерно каждые два (2) года, выпуск будет создаваться напрямую из main. Исторически такие выпуски назывались выпуском «точка-ноль (dot-zero)».

Эта статья освещает рабочий процесс и обязанности команды FreeBSD Release Engineering Team как для выпуска "точка-ноль", так и для "промежуточных" выпусков.

Следующие разделы этой статьи описывают:

Общая информация и подготовка

Общая информация и подготовка перед началом цикла выпуска.

Изменения на веб-сайте в течение цикла выпуска

Изменения на веб-сайте в течение цикла выпуска

Терминология выпуска релизов

Терминология и общая информация, такие как "мягкая заморозка кода (code slush)" и "заморозка кода (code freeze)", используемые в этом документе.

Выпуск релиза от main

Процесс Release Engineering для релиза "точка-ноль".

Выпуск релиза от stable/

Процесс Release Engineering для "точечного" выпуска.

Создание установочных носителей FreeBSD

Информация, относящаяся к конкретным процедурам создания установочного носителя.

Публикация установочных носителей FreeBSD на зеркалах проекта

Процедуры публикации установочного носителя.

Завершение цикла выпуска

Завершение цикла выпуска.

2. Общая информация и подготовка

Примерно за два месяца до начала цикла выпуска команда FreeBSD Release Engineering Team определяет график выпуска. График включает в себя различные этапы цикла выпуска, такие как даты заморозки, даты ветвления и даты сборки. Например:

Веха	Предполагаемая дата
мягкая заморозка main:	May 27, 2016
заморозка main:	June 10, 2016
заморозка main KBI:	June 24, 2016
мягкая заморозка дерева `doc/` [1]:	June 24, 2016
Ежеквартальная ветка Ports [2]:	July 1, 2016
Ветка stable/13:	July 8, 2016
`doc/` tree tag [3]:	July 8, 2016

Веха	Предполагаемая дата
Начало сборки BETA1:	July 8, 2016
Разморозка main:	July 9, 2016
Сборка BETA2 начинается:	July 15, 2016
Сборка BETA3 начинается [*]:	July 22, 2016
Ветка `releng/13.0`:	July 29, 2016
Сборка RC1 начинается:	July 29, 2016
Разморозка stable/13:	July 30, 2016
Сборка RC2 начинается:	August 5, 2016
Окончательные сборки пакетов Ports [4]:	August 6, 2016
Метка релиза портов:	12 августа 2016
Сборка RC3 начинается [*]:	12 августа 2016
Сборка RELEASE начинается:	August 19, 2016
Объявление о ВЫПУСКЕ:	September 2, 2016



Отметка "[*]" у элементов означает "по мере необходимости".

1. Мягкая заморозка дерева `doc/` координируется командой FreeBSD Documentation Engineering Team.
2. Используемая квартальная ветка Ports определяется по дате запланированной окончательной сборки `RC`. Новая квартальная ветка создаётся в первый день квартала, поэтому этот показатель следует учитывать, принимая во внимание этапы цикла выпуска. Квартальная ветка создаётся командой FreeBSD Ports Management Team.
3. Для дерева исходного кода `doc/` tag делается командой FreeBSD Documentation Engineering Team.
4. Окончательная сборка пакета Ports выполняется командой FreeBSD Ports Management Team после финальной (или ожидаемой финальной) сборки `RC`.



Если выпуск создаётся из существующей ветки stable/, дату заморозки KBI можно исключить, так как KBI уже считается замороженным в устоявшихся ветках stable/.

При написании графика цикла выпуска необходимо учитывать ряд факторов, особенно этапы, на которые целевая дата зависит от предопределённых этапов, от которых существует зависимость. Например, тег выпуска коллекции портов берется из активной квартальной ветки на момент последнего `RC`. Это частично определяет, какая квартальная ветка используется, когда может быть создан тег выпуска и какая версия дерева портов используется для финальной сборки `RELEASE`.

После общего согласования расписания команда FreeBSD Release Engineering Team отправляет расписание разработчикам FreeBSD по электронной почте.

Вполне типично, что многие разработчики сообщают команде FreeBSD Release Engineering Team о различных работах в процессе выполнения. В некоторых случаях может быть запрошено продление для работы в процессе, а в других случаях может быть сделан запрос на "общее одобрение" для определённого подмножества дерева.

При таких запросах важно убедиться, что обсуждаются сроки (даже если они приблизительные). Для общих одобрений следует чётко указать продолжительность действия такого одобрения. Например, разработчик FreeBSD может запросить общее одобрение с начала периода мягкой заморозки кода до начала сборок RC.



Для отслеживания общих одобрений команда FreeBSD Release Engineering Team использует внутренний репозиторий, в котором ведется журнал таких запросов. В нём указывается область, на которую распространяется общее одобрение, автор(ы), срок действия одобрения и причина его выдачи. Например, может быть предоставлено общее одобрение для release/doc/ всем членам FreeBSD Release Engineering Team до финального RC для обновления примечаний к выпуску и другой связанной с выпуском документации.



Команда FreeBSD Release Engineering Team также использует этот репозиторий для отслеживания ожидающих одобрения запросов, полученных непосредственно перед началом различных сборок во время цикла выпуска, при этом Инженер по выпускам указывает срок отсечки с помощью электронного письма разработчикам FreeBSD.

В зависимости от рассматриваемого набора кода и общего влияния этого набора кода на FreeBSD в целом, такие запросы могут быть одобрены или отклонены командой FreeBSD Release Engineering Team.

То же самое относится к расширениям для незавершённых работ. Например, незавершённая работа над новым драйвером устройства, который в остальном изолирован от остального дерева, может получить расширение. Однако новый планировщик может оказаться неосуществимым, особенно если такие значительные изменения отсутствуют в другой ветке.

Расписание также добавляется на сайт проекта в репозитории doc/, в файле ~/website/content/en/releases/13.0R/schedule.adoc. Этот файл постоянно обновляется по мере прохождения цикла выпуска.



В большинстве случаев файл schedule.adoc можно скопировать из предыдущего релиза и обновить соответствующим образом.

В дополнение к добавлению schedule.adoc на веб-сайт, файл ~/shared/releases.adoc также обновляется, чтобы добавить ссылку на расписание на различные подстраницы, а также включить ссылку на расписание на главной странице веб-сайта проекта.

Расписание также доступно по ссылке из файла ~/website/content/en/releng/_index.adoc.

Примерно за месяц до запланированной "мягкой заморозки кода" команда FreeBSD Release

Engineering Team отправляет напоминание разработчикам FreeBSD.

3. Терминология выпуска релизов

В этом разделе описана часть терминологии, используемой в остальной части данного документа.

3.1. Мягкая заморозка кода

Хотя мягкая заморозка кода не является полной заморозкой дерева, команда FreeBSD Release Engineering Team просит отдавать приоритет исправлению ошибок в существующей кодовой базе перед добавлением новых функций.

Мягкая заморозка кода не требует подтверждений коммитов в ветку.

3.2. Заморозка кода

Замораживание кода означает момент времени, когда все коммиты в ветку требуют явного одобрения от FreeBSD Release Engineering Team.

Репозиторий FreeBSD Git содержит несколько хуков для выполнения проверок перед тем, как изменения будут зафиксированы в дереве. Один из этих хуков проверяет, требуются ли специальные разрешения для фиксации изменений в определённой ветке.

Для обеспечения утверждения коммитов командой FreeBSD Release Engineering Team, команда Release Engineering должна утверждать любые изменения в ветке. В этом случае журнал коммитов должен содержать строку `Approved by: re (логин)`, где "логин" — это идентификатор утверждающего.



Во время заморозки кода участники проекта FreeBSD должны следовать Рекомендациям по запросам изменений.

3.3. Замораживание KBI/KPI

KBI/KPI стабильность подразумевает, что вызов функции в двух разных релизах программного обеспечения, реализующего эту функцию, приводит к одинаковому конечному состоянию. Вызывающая сторона, будь то процесс, поток или функция, ожидает, что функция будет работать определённым образом, в противном случае стабильность KBI/KPI на ветке нарушается.

4. Изменения на веб-сайте в течение цикла выпуска

Этот раздел описывает изменения на веб-сайте, которые должны происходить по мере развития цикла выпуска.



Файлы, указанные в этом разделе, относятся к ветке `main` репозитория `doc`.

4.1. Изменения на веб-сайте перед началом цикла выпуска

Когда график цикла выпуска становится доступным, эти файлы необходимо обновить, чтобы включить различные функции на веб-сайте проекта FreeBSD:

Файл для редактирования	Что изменить
~/shared/releases.adoc	Изменить `beta-upcoming` с `IGNORE` на `INCLUDE`
~/shared/releases.adoc	Изменить `beta-testing` с `IGNORE` на `INCLUDE`

4.2. Изменения на веб-сайте в период BETA или RC

При переходе от `PRERELEASE` к `BETA` эти файлы необходимо обновить, чтобы включить блок "Help Test (Помогите протестировать)" на странице загрузки. Все файлы указаны относительно head/ в репозитории `doc`:

Файл для редактирования	Что изменить
~/shared/releases.adoc	Обновите `betarel-vers` до `BETA1`
~/website/data/en/news/news.toml	Добавьте запись, объявляющую о `BETA`
~/website/static/security/advisory-template.txt	Добавьте новую `BETA`, `RC` или финальную `RELEASE` в шаблон
~/website/static/security/errata-template.txt	Добавьте новую `BETA`, `RC` или финальную `RELEASE` в шаблон

После создания ветки releng/13.0 необходимо добавить различные документы, связанные с выпуском, в репозиторий `doc/`.



Соответствующие документы, связанные с выпусками, находятся в репозитории doc для FreeBSD 12.x и более поздних версий.

4.3. Изменения в портах во время BETA, RC и финального RELEASE

Для каждой сборки в течение цикла выпуска файлы `MANIFEST`, содержащие `SHA256` для различных дистрибутивных наборов, таких как `base.txz`, `kernel.txz` и других, добавляются в порт misc/freebsd-release-manifests. Это позволяет другим утилитам, помимо bsdinstall(8), таким как ports-mgmt/poudriere, безопасно использовать эти дистрибутивные наборы, предоставляя механизм для проверки контрольных сумм.

5. Выпуск релиза от main

Этот раздел описывает общие процедуры цикла выпуска FreeBSD из ветки main.

5.1. FreeBSD “ALPHA” сборки

Начиная с цикла выпуска FreeBSD 10.0-RELEASE, было введено понятие сборок “ALPHA”. В отличие от сборок BETA и RC, сборки ALPHA не включены в график выпуска FreeBSD.

Идея сборок ALPHA заключается в предоставлении регулярных сборок FreeBSD до создания ветки stable/.

Снимки состояния ALPHA FreeBSD должны собираться примерно раз в неделю.

Для первой сборки ALPHA значение BRANCH в sys/conf/newvers.sh необходимо изменить с CURRENT на ALPHA1. Для последующих сборок ALPHA увеличивайте каждое значение ALPHAN на единицу.

См. Сборка установочных носителей FreeBSD для получения информации о сборке образов ALPHA.

5.2. Создание ветки stable/13

При создании ветки stable/ необходимо внести несколько изменений как в новой ветке stable/, так и в ветке main. Указанные файлы относятся к корню репозитория. Чтобы создать новую ветку stable/13 в Git:



Убедитесь, что вы находитесь в ветке main

```
% git checkout -b stable/13
```

После создания ветки stable/13 внесите следующие изменения:

Файл для редактирования	**Что изменить**
UPDATING	Обновите версию FreeBSD и удалите уведомление о WITNESS

contrib/jemalloc/include/jemalloc/jemalloc_FreeBSD.h	`#ifndef MALLOC_PRODUCTION`
	`#define MALLOC_PRODUCTION`
	`#endif`

lib/clang/llvm.build.mk	Раскомментируйте -DNDEBUG
sys/*/conf/GENERIC*	Удалите поддержку отладки
sys/*/conf/MINIMAL	Удалите поддержку отладки
release/release.conf.sample	Обновите SRCBRANCH

Файл для редактирования	Что изменить
sys/*/conf/GENERIC-NODEBUG	Удалите следующие параметры конфигурации ядра
sys/arm/conf/std.arm*	Удалите отладочные опции
sys/conf/newvers.sh	Обновите значение `BRANCH`, чтобы оно соответствовало `BETA1`
share/mk/src.opts.mk	Переместите `REPRODUCIBLE_BUILD` из `__DEFAULT_NO_OPTIONS` в `__DEFAULT_YES_OPTIONS`
share/mk/src.opts.mk	Переместите `LLVM_ASSERTIONS` из `__DEFAULT_YES_OPTIONS` в `__DEFAULT_NO_OPTIONS`
libexec/rc/rc.conf	Установите `dumpdev` с `AUTO` на `NO` (это настраивается для тех, кто хочет включить его по умолчанию)
release/Makefile	Удалите записи `debug.witness.trace`

Затем в ветке main, которая теперь станет новой основной версией:

Файл для редактирования	Что изменить
UPDATING	Обновите версию FreeBSD
sys/conf/newvers.sh	Обновите значение `BRANCH`, чтобы оно соответствовало `CURRENT`, и увеличьте `REVISION`
Makefile.inc1	Обновите `TARGET_TRIPLE` и `MACHINE_TRIPLE`
sys/sys/param.h	Обновите `__FreeBSD_version`
gnu/usr.bin/cc/cc_tools/freebsd-native.h	Обновите `FBSD_MAJOR` и `FBSD_CC_VER`
contrib/gcc/config.gcc	Добавьте раздел `freebsdversion.h`
lib/clang/llvm.build.mk	Обновите значение `OS_VERSION`
lib/clang/freebsd_cc_version.h	Обновите `FREEBSD_CC_VERSION`
lib/clang/include/lld/Common/Version.inc	Обновите `LLD_REVISION_STRING`
Makefile.libcompat	Обновите `LIB32CPUFLAGS`

6. Выпуск релиза от stable/

Этот раздел описывает общие процедуры цикла выпуска FreeBSD из установленной ветки stable/.

6.1. Мягкая заморозка кода ветки `stable` FreeBSD

В рамках подготовки к заморозке кода в ветке `stable` необходимо обновить несколько файлов, чтобы отразить официальное начало цикла выпуска. Эти файлы находятся в корневом каталоге ветки stable:

Файл для редактирования	Что изменить
sys/conf/newvers.sh	Обновите значение BRANCH, чтобы отразить PRERELEASE
Makefile.inc1	Обновите TARGET_TRIPLE
lib/clang/llvm.build.mk	Обновите OS_VERSION
Makefile.libcompat	Обновите LIB32CPUFLAGS

6.2. Сборки FreeBSD BETA

После периода мягкой заморозки следующая фаза цикла выпуска — это заморозка кода. На этом этапе все коммиты в стабильную ветку требуют явного одобрения от FreeBSD Release Engineering Team. Это контролируется gitadm@FreeBSD.org, который управляет репозиторием.



Существует два общих исключения, когда не требуется подтверждение коммита во время цикла выпуска. Первое — любые изменения, которые необходимо закоммитить инженеру выпуска для продолжения ежедневной работы цикла выпуска. Второе — исправления безопасности, которые могут возникнуть во время цикла выпуска.

После вступления в силу заморозки кода следующая сборка из ветки помечается как BETA1. Это делается путём изменения значения BRANCH в файле sys/conf/newvers.sh с PRERELEASE на BETA1.

После этого начинается первая сборка BETA. Последующие сборки BETA не требуют обновления каких-либо файлов, кроме sys/conf/newvers.sh, с увеличением номера сборки BETA.

6.3. Создание ветки releng/13.0

Когда первая сборка RC (Release Candidate) готова к началу, создаётся ветка releng/. Это многоэтапный процесс, который должен выполняться в определённом порядке, чтобы избежать аномалий, таких как пересечения значений __FreeBSD_version, например. Указанные ниже пути относятся к корню репозитория. Порядок коммитов и что нужно изменить:



Убедитесь, что вы находитесь в ветке stable/13

```
% git checkout -b releng/13.0
```

Файл для редактирования	Что изменить
sys/conf/newvers.sh	Измените BETAX на RC1
sys/sys/param.h	Обновите __FreeBSD_version

Файл для редактирования	Что изменить
sys/conf/kern.opts.mk	Переместите REPRODUCIBLE_BUILD из DEFAULT_NO_OPTIONS вDEFAULT_YES_OPTIONS
etc/pkg/FreeBSD.conf	Замените latest на quarterly в качестве расположения репозитория пакетов по умолчанию
release/pkg_repos/release-dvd.conf	Замените latest на quarterly в качестве расположения репозитория пакетов по умолчанию
sys/conf/newvers.sh	Обновите BETAX на PRERELEASE
sys/sys/param.h	Обновите __FreeBSD_version

Затем gitadm@FreeBSD.org добавляет новых утверждающих для ветки releng, как это было сделано для ветки stable.

```
% git add .
% git commit
```

Теперь, когда существуют два новых значения __FreeBSD_version, также обновите файл ~/documentation/content/en/books/porters-handbook/versions/chapter.adoc в репозитории проекта документации.

После завершения первой сборки RC и её тестирования ветку stable/ можно «разморозить» с помощью gitadm@FreeBSD.org.

После появления первой версии RC необходимо отправить письмо команде FreeBSD Bugmeister Team, чтобы добавить новую версию FreeBSD -RELEASE в список versions, доступный в выпадающем меню трекера ошибок.

7. Создание установочных носителей FreeBSD

Этот раздел описывает общие процедуры создания снимков разработки и выпусков FreeBSD.

7.1. Скрипты сборки релизов

До выхода FreeBSD 9.0-RELEASE файл src/release/Makefile был обновлен для поддержки bsdinstall(8), а скрипт src/release/generate-release.sh был добавлен в качестве обёртки для автоматизации вызова целей release(7).

До выхода FreeBSD 9.2-RELEASE был представлен src/release/release.sh, который, основываясь на src/release/generate-release.sh, включал поддержку указания конфигурационных файлов для переопределения различных опций и переменных окружения. Поддержка

конфигурационных файлов обеспечила возможность кросс-сборки каждого архитектурного варианта для релиза путем указания отдельного конфигурационного файла для каждого вызова.

В качестве краткого примера использования src/release/release.sh для сборки одного релиза в /scratch:

```
# /bin/sh /usr/src/release/release.sh
```

В качестве краткого примера использования src/release/release.sh для сборки единого кросс-собранного выпуска с использованием другого целевого каталога, создайте пользовательский release.conf, содержащий:

```
# release.sh configuration for powerpc/powerpc64
CHROOTDIR="/scratch-powerpc64"
TARGET="powerpc"
TARGET_ARCH="powerpc64"
KERNEL="GENERIC64"
```

Затем выполните src/release/release.sh следующим образом:

```
# /bin/sh /usr/src/release/release.sh -c $HOME/release.conf
```

См. release(7) и src/release/release.conf.sample для получения дополнительных сведений и примеров использования.

7.2. Сборка релизов FreeBSD

В течение цикла выпуска копии файлов CHECKSUM.SHA512 и CHECKSUM.SHA256 для каждой архитектуры сохраняются во внутреннем репозитории FreeBSD Release Engineering Team, а также включаются в различные рассылки с объявлениями. Каждый файл MANIFEST, содержащий хеши base.txz, kernel.txz и других, также добавляется в пакет misc/freebsd-release-manifests в Коллекции портов.

В подготовке к сборке выпуска необходимо обновить несколько файлов:

Файл для редактирования	**Что изменить**
sys/conf/newvers.sh	Обновите значение `BRANCH` на `RELEASE`
UPDATING	Добавьте предполагаемую дату объявления
lib/csu/common/crtbrand.S	Замените `__FreeBSD_version` на значение из sys/sys/param.h

После сборки окончательного `RELEASE`, ветка releng/13.0 помечается как release/13.0.0, используя ревизию, из которой был собран `RELEASE`. Аналогично созданию веток stable/13 и

releng/13.0, это делается с помощью `git tag`. Из корня репозитория:



Убедитесь, что вы находитесь в ветке releng/13.0

```
% git tag release/13.0.0
```

8. Публикация установочных носителей FreeBSD на зеркалах проекта

Этот раздел описывает процедуру публикации снимков разработки FreeBSD и выпусков на зеркала Проекта.

8.1. Подготовка образов установочных носителей FreeBSD

Этап подготовки (staging) снимков и выпусков FreeBSD — это двухэтапный процесс:

- Создание структуры каталогов, соответствующей иерархии на `ftp-master`

  Если `EVERYTHINGISFINE` определено в конфигурационных файлах сборки, main.conf в случае скриптов сборки, упомянутых выше, это происходит автоматически в chroot(8) после завершения сборки, создавая структуру каталогов в ${DESTDIR}/R/ftp-stage с путями, соответствующими ожидаемым на `ftp-master`. Это эквивалентно выполнению следующего в chroot(8):

  ```
  # make -C /usr/src/release -f Makefile.mirrors EVERYTHINGISFINE=1 ftp-stage
  ```

  После сборки каждой архитектуры скрипт thermite.sh выполнит синхронизацию ${DESTDIR}/R/ftp-stage с билда в каталоге /snap/ftp/snapshots или /snap/ftp/releases на хосте сборки, соответственно.

- Копирование файлов в промежуточный каталог на `ftp-master` перед перемещением файлов в pub/ для начала распространения на зеркала Проекта

  После завершения всех сборок каталог /snap/ftp/snapshots (или /snap/ftp/releases для релиза) опрашивается `ftp-master`-ом с использованием rsync для передачи в /archive/tmp/snapshots или /archive/tmp/releases, соответственно.

  

  На `ftp-master` в инфраструктуре проекта FreeBSD этот шаг требует доступа уровня `root`, так как он должен выполняться от имени пользователя `archive`.

8.2. Публикация установочных носителей FreeBSD

Как только образы размещены в /archive/tmp/, они готовы к публикации путем размещения в /archive/pub/FreeBSD. Для уменьшения времени распространения используется pax(1) для создания жёстких ссылкок из /archive/tmp в /archive/pub/FreeBSD.



Для эффективной работы и /archive/tmp, и /archive/pub должны находиться в одной логической файловой системе.

Однако есть оговорка: после этого необходимо использовать rsync после pax(1) для исправления символических ссылок в pub/FreeBSD/snapshots/ISO-IMAGES, которые pax(1) заменит на жёсткие ссылки, что увеличит время распространения.



Как и на этапе подготовки, это требует доступа уровня `root`, так как данный шаг должен быть выполнен от имени пользователя `archive`.

Как пользователь `archive`:

```
% cd /archive/tmp/snapshots
% pax -r -w -l . /archive/pub/FreeBSD/snapshots
% /usr/local/bin/rsync -avH /archive/tmp/snapshots/* /archive/pub/FreeBSD/snapshots/
```

Замените *снимки* на *релизы* там, где это уместно.

9. Завершение цикла выпуска

Этот раздел описывает общие задачи после выпуска.

9.1. Уведомления об исправлениях после выпуска

По мере приближения цикла выпуска к завершению часто появляются несколько кандидатов в EN (Errata Notice — уведомления об ошибках) для устранения проблем, обнаруженных на поздних этапах цикла. После выпуска FreeBSD Release Engineering Team и FreeBSD Security Team пересматривают изменения, которые не были одобрены до финального выпуска, и, в зависимости от масштаба рассматриваемого изменения, могут выпустить EN.



Фактический процесс выпуска EN обрабатывается командой FreeBSD Security Team.

Для запроса уведомления об ошибке после завершения цикла выпуска разработчик должен заполнить шаблон уведомления об ошибке, в частности разделы `Предыстория`, `Описание проблемы`, `Последствия` и, если применимо, `Обходное решение`.

Заполненный шаблон уведомления об ошибке следует отправить по электронной почте вместе с патчем для ветки releng/ или списком ревизий из ветки stable/.

Для запросов на уведомления об ошибках (Errata Notice), поступающих сразу после выпуска, запрос следует отправлять по электронной почте как в FreeBSD Release Engineering Team, так и в FreeBSD Security Team. После того как ветка releng/ передана FreeBSD Security Team, как описано в Передача FreeBSD Security Team, запросы на уведомления об ошибках следует направлять в FreeBSD Security Team.

9.2. Передача в FreeBSD Security Team

Примерно через две недели после выпуска релиза инженер по релизам обновляет репозиторий Git, изменяя утверждающего с команды инженеров по релизам на офицера безопасности для ветки `releng/13.0`.

10. Конец срока поддержки выпуска

Этот раздел описывает файлы, связанные с веб-сайтом, которые необходимо обновить, когда выпуск достигает EoL (End-of-Life).

10.1. Обновления веб-сайта для прекращения поддержки

Когда выпуск достигает конца жизненного цикла, ссылки на этот выпуск должны быть удалены и/или обновлены на веб-сайте:

Файл	Что изменить
~/website/themes/beastie/layouts/index.html	Удалите ссылки на `u-relXXX-announce` и `u-relXXX-announce`.
~/website/content/en/releases/_index.adoc	Переместите переменные `u-relXXX-*` из списка поддерживаемых выпусков в список Устаревших выпусков.
~/website/content/en/releng/_index.adoc	Обновите соответствующую ветку releng, чтобы отразить, что ветка больше не поддерживается.
~/website/content/en/security/_index.adoc	Удалить ветку из списка поддерживаемых веток.
~/website/content/en/security/unsupported.adoc	Добавить ветку в список неподдерживаемых веток.
~/website/content/en/where.adoc	Удалите URL-адреса для выпуска.
~/website/themes/beastie/layouts/partials/sidenav.html	Удалите ссылки на `u-relXXX-announce` и `u-relXXX-announce`.
~/website/static/security/advisory-template.txt	Удалите ссылки на ветку release и releng.
~/website/static/security/errata-template.txt	Удалите ссылки на ветку release и releng.