

Генерация Debian changelog

Для автоматического формирования файла `debian/changelog` используется скрипт `scripts/generate-changelog.sh`.

Скрипт предназначен для подготовки Debian-пакета и формирует changelog на основании истории Git. Версия пакета берётся из `CMakeLists.txt`, а список изменений — из коммитов ветки `develop`, отсутствующих в ветке `main`.

```
#!/bin/bash

set -euo pipefail

PROJECT_NAME="pingfinder-msgd"
DEBIAN_REVISION="1"

# Get the repository root
REPO_ROOT=$(git rev-parse --show-toplevel)

CMAKE_FILE="${REPO_ROOT}/CMakeLists.txt"
CHANGELOG_FILE="${REPO_ROOT}/debian/changelog"

# Check current branch
CURRENT_BRANCH=$(git branch --show-current)

if [[ "$CURRENT_BRANCH" != "develop" ]]; then
    echo "Error: script must be run on the develop branch." >&2
    exit 1
fi

# Extract project version from CMakeLists.txt
VERSION=$(sed -nE \
"s/^project\(${PROJECT_NAME}[:space:]+\)+VERSION[:space:]+\([0-9]+\.[0-9]+\.[0-9]+\)+\./\1/p" \
"$CMAKE_FILE")

if [[ -z "$VERSION" ]]; then
    echo "Error: project version not found in $CMAKE_FILE." >&2
    exit 1
fi

echo "Project version: $VERSION"

# Find the latest release tag
LATEST_TAG=$(git tag --list 'v[0-9]*.[0-9]*.[0-9]*' --sort=-version:refname | head -n 1)

if [[ -n "$LATEST_TAG" ]]; then
    LATEST_VERSION="${LATEST_TAG#v}"

    echo "Latest release: $LATEST_TAG"
```

```
if dpkg --compare-versions "$VERSION" lt "$LATEST_VERSION"; then
    echo "Error: project version $VERSION is older than latest release
$LATEST_VERSION." >&2
    exit 1
elif dpkg --compare-versions "$VERSION" eq "$LATEST_VERSION"; then
    echo "Version status: current release version."
else
    echo "Version status: new release version."
fi
else
    LATEST_VERSION=""
    echo "Latest release: none"
fi

# Find the latest Debian changelog update commit.
# This commit is the release point in develop.
RELEASE_POINT=$(git log -1 \
    --format='%H' \
    --grep='^Update Debian changelog for ')

if [[ -n "$RELEASE_POINT" ]]; then
    echo "Latest release point: $(git show -s --format='%s' "$RELEASE_POINT")"
else
    echo "Latest release point: none"
fi

# Get commits accumulated since the latest release point.
if [[ -n "$RELEASE_POINT" ]]; then
    LOG_RANGE="${RELEASE_POINT}..develop"
else
    LOG_RANGE="develop"
fi

COMMITTS=$(git log "$LOG_RANGE" \
    --pretty=format: '%s' \
    --reverse \
    --grep='^Update Debian changelog for' \
    --invert-grep)

# Find the latest release preparation commit for the current version.
# It determines the date used in the Debian changelog.
RELEASE_COMMIT=$(git log -1 \
    --format='%H' \
    --grep="^Prepare release ${VERSION}$")

if [[ -z "$RELEASE_COMMIT" ]]; then
    echo "Error: release preparation commit not found for version $VERSION." >&2
    exit 1
fi
```

```
CHANGELOG_DATE=$(git show -s --format=%aD "$RELEASE_COMMIT")

# Handle absence of new commits
if [[ -z "$COMMIT" ]]; then
    if [[ -n "$LATEST_TAG" ]] && \
        dpkg --compare-versions "$VERSION" eq "$LATEST_VERSION"; then
        echo "No changes since the latest release."
    else
        echo "Error: no changes found since the previous release." >&2
        exit 1
    fi
fi

# Get maintainer information from Git configuration
MAINTAINER_NAME=$(git config user.name)
MAINTAINER_EMAIL=$(git config user.email)

if [[ -z "$MAINTAINER_NAME" || -z "$MAINTAINER_EMAIL" ]]; then
    echo "Error: Git user.name or user.email is not configured." >&2
    exit 1
fi

# Generate Debian changelog
{
    echo "$PROJECT_NAME (${VERSION}-${DEBIAN_REVISION}) unstable; urgency=medium"
    echo

    if [[ -n "$COMMIT" ]]; then
        while IFS= read -r commit; do
            echo "    * $commit"
        done <<< "$COMMIT"
    fi

    echo
    printf ' -- %s <%s> %s\n' \
        "$MAINTAINER_NAME" \
        "$MAINTAINER_EMAIL" \
        "$CHANGELOG_DATE"
} > "$CHANGELOG_FILE"

echo "Generated $CHANGELOG_FILE:"
echo
cat "$CHANGELOG_FILE"
```

Условия запуска

Скрипт необходимо запускать из Git-репозитория проекта. Текущей веткой должна быть develop:

```
./generate-changelog.sh
```

Скрипт самостоятельно определяет корень репозитория, поэтому его можно запускать непосредственно из каталога `scripts`.

Для работы скрипта необходимы:

- `git`;
- `dpkg`, предоставляющий утилиту `dpkg --compare-versions`;
- корректно настроенные параметры `Git user.name` и `user.email`.

Версия проекта

Версия проекта определяется из строки `project()` файла `CMakeLists.txt`:

```
project(pingfinder-msgd VERSION 0.0.3)
```

Таким образом, `CMakeLists.txt` является единственным источником версии проекта. Версия в Debian-файлах отдельно не задаётся.

Скрипт также определяет последний `release tag` вида:

```
v<major>.<minor>.<patch>
```

и сравнивает его версию с версией из `CMakeLists.txt`.

Возможны три состояния:

- версия совпадает с последним релизом — нормальное состояние в процессе разработки;
- версия выше последнего релиза — проект подготавливается к новому релизу;
- версия ниже последнего релиза — ошибка, поскольку версия проекта не может быть старше уже выпущенной.

Поэтому изменение версии в `CMakeLists.txt` выполняется непосредственно при подготовке нового релиза. Во время обычной разработки версия может совпадать с версией последнего выпущенного релиза.

Формирование списка изменений

В `changelog` попадают коммиты, которые присутствуют в `develop`, после предыдущего коммита `Update Debian changelog for X.Y.Z`, который выступает в роли `Release point` в ветке `develop`.

Принята следующая модель ветвления:

- `main` — последний выпущенный код без истории коммитов;
- `develop` — накопленные изменения с предыдущего релиза.

Сообщения коммитов сохраняются в исходном порядке и преобразуются в записи `Debian changelog`:

- `Commit message`

Предполагается, что сообщения коммитов должны быть достаточно информативными и понятными без дополнительного пояснения.

Формирование Debian changelog

На основании полученной информации создаётся файл: `debian/changelog`

Его заголовок имеет следующий вид:

```
pingfinder-msgd (0.0.3-1) unstable; urgency=medium
```

Здесь:

- 0.0.3 — версия проекта из CMakeLists.txt;
- 1 — Debian revision, задаваемый в `generate-changelog.sh` в переменной `DEBIAN_REVISION`;
- unstable — Debian distribution;
- medium — приоритет изменений.

В конце записи указываются имя и email разработчика из настроек Git:

- `git config user.name`
- `git config user.email,`

а также текущая дата в формате RFC 2822.

From:

<https://jurik-phys.net.ru/> - **Jurik-Phys.Net.Ru**

Permanent link:

<https://jurik-phys.net.ru/itechnology:linux-pkg-deb:generate-changelog>

Last update: **2026/09/29 02:01**

