



# **Сведения о хранении исходного кода и средствах компиляции**

**Laduga**

**июн. 30, 2026**



<b>1</b>	<b>Хранение исходного кода, объектного кода и средства сборки</b>	<b>1</b>
1.1	Назначение документа . . . . .	1
1.2	Хранение исходного кода . . . . .	1
1.3	Хранение объектного кода и дистрибутивов . . . . .	2
1.4	Средства компиляции . . . . .	2
1.5	Использование Qt . . . . .	3
1.6	СИ-сборка и публикация артефактов . . . . .	3
1.7	Хранение лицензионных ключей и секретов . . . . .	3



---

## Хранение исходного кода, объектного кода и средства сборки

---

### 1.1 Назначение документа

Настоящий раздел содержит описание технических средств хранения исходного текста и объектного кода программного обеспечения PRADIS, а также технических средств компиляции исходного текста в объектный код.

### 1.2 Хранение исходного кода

Исходный код PRADIS хранится во внутренней системе управления версиями GitLab на собственных серверах организации, расположенных на территории Российской Федерации. Доступ к репозиториям предоставляется уполномоченным сотрудникам и регулируется средствами разграничения прав GitLab.

В состав исходных кодов и сборочной инфраструктуры входят следующие основные репозитории:

Репозиторий	Назначение
pradis/pradis-setup	Сценарии CI, сборка установочных пакетов, подготовка дистрибутивов и публикация артефактов.
pradis/pq/pq	Исходный код графической среды PRADIS/Qucs на C++ с использованием Qt.
pradis/post/post	Исходный код постпроцессора на C++ с использованием Qt и Open CASCADE.
pradis/modules/basis	Базовые модули и артефакты PRADIS. Штатная CI-процедура формирует архив артефактов; legacy-компиляция может включаться отдельно переменной BASIS_COMPILE_LEGACY.

### 1.3 Хранение объектного кода и дистрибутивов

Объектный код, исполняемые файлы, библиотеки и установочные пакеты формируются в процессе CI-сборки. Результаты сборки сохраняются как артефакты GitLab CI и, для релизных или кандидатных сборок, публикуются во внутреннее Nexus-хранилище организации.

К объектному коду и сборочным артефактам относятся:

- исполняемые файлы и динамические библиотеки компонентов PRADIS;
- промежуточные и итоговые артефакты CI-сборки;
- установочные пакеты для Linux-дистрибутивов, включая .rpm и .deb;
- архивы компонентных payload-артефактов, используемые при формировании установочного пакета.

Доступ к объектному коду, пакетам и архивам регулируется правами доступа в GitLab CI и Nexus.

### 1.4 Средства компиляции

Программное обеспечение PRADIS содержит компилируемые компоненты, написанные преимущественно на C и C++. Для подготовки сборки, вспомогательных операций и упаковки также используются сценарии Bash, PowerShell и Python.

Компиляция исходного кода в объектный код для Linux-сборок выполняется на серверах GitLab Runner, расположенных на территории Российской Федерации. Сборка выполняется в контролируемом CI-окружении с использованием Docker образов целевых Linux-дистрибутивов.

В качестве технических средств компиляции и сборки применяются следующие программные продукты:

Средство	Назначение
gcc / g++	Компиляторы GNU для компиляции исходного кода C и C++ в объектный код.
cmake	Кроссплатформенная система конфигурирования и автоматизации сборки проектов.
ninja-build	Высокопроизводительный исполнитель сборочных задач, используемый совместно с CMake.
make	Исполнитель сборочных сценариев и правил сборки для отдельных компонентов.
Qt5 development packages	Библиотеки и средства разработки Qt5, используемые при сборке графических компонентов PRADIS.
rpm-build и dpkg-dev	Средства формирования установочных пакетов .rpm и .deb.

## 1.5 Использование Qt

В Linux-сборке применяются библиотеки и средства разработки Qt5. В составе исходного кода присутствует модуль `qt3_compat`, реализующий совместимость с устаревшими Qt3-подобными API. Данный модуль является частью исходного кода PRADIS и собирается вместе с проектом; отдельный Qt3 toolchain для Linux-сборки не используется.

## 1.6 CI-сборка и публикация артефактов

CI-процессы выполняются во внутреннем GitLab. Основные стадии включают сборку, проверку установки и публикацию артефактов. Для Linux-сборок используются задания, формирующие пакеты для Astra Linux, РЕД ОС и ALT Linux.

Сборочные задания получают исходные коды из внутренних Git-репозиториев, компилируют компоненты, устанавливают их во временный корень пакета, формируют установочный пакет и сохраняют результат как GitLab CI artifacts. При наличии настроенных учетных данных результат дополнительно публикуется во внутреннее Nexus-хранилище.

## 1.7 Хранение лицензионных ключей и секретов

Лицензионные ключи, токены доступа и пароли не хранятся в исходном коде и не публикуются в документации. Для CI-сборок используются защищенные переменные GitLab CI/CD и иные средства хранения секретов с ограниченным доступом.

В документации указывается только порядок хранения и использования лицензионных данных без раскрытия значений ключей, токенов и паролей.