

Инструкция по эксплуатации программного обеспечения FlexSoft Java Development Platform (FlexSoft JDP)

Список компонент FlexSoft JDP	3
Конвертер кода - PLSQLJAVA Utils	4
1. Назначение утилиты	4
2. Состав поставки	4
3. Режимы работы	4
3.1. Режим java	4
3.2. Режим plsql-feature-check	4
3.3. Режим validate-sql	4
4. Требования к окружению	5
5. Технические таблицы	5
6. Вспомогательные таблицы	5
7. Первый запуск	6
8. Параметры запуска	7
9.1. Основные параметры	7
9.2. Дополнительные параметры для validate-sql	7
9. Когда использовать RECREATE_SCHEMA=true	7
10. Рекомендуемый способ запуска	8
11. Примеры запуска	8
11.1. Режим java	8
11.2. Режим plsql-feature-check	8
11.3. Режим validate-sql	9
12. Формат результатов режима validate-sql	9
12.1. Файлы ошибок	9
12.2. Сводный файл	9
12.3. Структура набора данных по одной ошибке	9
12.4. Поиск исходного запроса	10
13. Замечания	10
Модуль миграции Database Migration Utility	11
1. Назначение	11
2. Подготовка	11
3. Конфигурация (БД-источник)	11
4. Запуск	11
5. Требования к окружению	12
Серверный рантайм – Data Access Framework (DAF)	13
1. Порядок запуска	13
2. Особенности запуска	13

Список компонент FlexSoft JDP

Конвертер кода - PLSQLJAVA Utils

Модуль миграции данных - Database Migration Utility

Серверный рантайм - Data Access Framework (DAF)

Конвертер кода - PLSQLJAVA Utils

Транспиляция PL/SQL → Java | Проверка конструкций | Валидация SQL на Platform V Datagrid

1. Назначение утилиты

plsql-transpiler-tool — утилита для анализа PL/SQL-кода в Oracle и работы с ним в одном из трёх режимов:

Режим	Описание
java	Транспиляция PL/SQL-объектов в Java-код
plsql-feature-check	Проверка полноты поддержки PL/SQL-конструкций для конвертации
validate-sql	Валидация SQL-запросов на Platform V Datagrid

Результаты записываются на диск в каталог, указанный параметром BASE_DIRECTORY.

2. Состав поставки

- plsql-transpiler-tool-1.0.jar
- plsql-transpiler-tool.bat, plsql-transpiler-tool.sh
- настоящая инструкция

3. Режимы работы

3.1. Режим java

Транспиляция PL/SQL-кода в Java.

Результат работы:

- набор Java-классов
- сгенерированные каталоги по значению DEFAULT_JAVA_PACKAGE
- минимальный набор Java-объектов, необходимый для запуска кода

3.2. Режим plsql-feature-check

Проверка полноты поддержки PL/SQL-конструкций.

Результат работы:

- отчёты о конструкциях без реализации конвертации
- служебные логи и статистика (если предусмотрены конфигурацией)

3.3. Режим validate-sql

Валидация SQL-запросов на Ignite.

В этом режиме утилита:

- извлекает SQL-конструкции из Oracle
- проверяет их выполнимость на Ignite
- формирует файлы с ошибками и итоговую статистику

Важно: Дополнительно требуется передать параметры подключения к Platform V Datagrid.

Результат работы:

- файлы с описанием ошибок по отдельным объектам
- сводный файл статистики summary.txt

4. Требования к окружению

- Windows с возможностью запуска .bat ИЛИ Linux с возможностью запуска sh (sh||bash)
- JDK 21
- jar-файл утилиты и bat||sh-файл в одном каталоге
- доступ к Oracle
- для режима validate-sql — доступ к Platform V Datagrid, завершённая миграция таблиц

5. Технические таблицы

Таблицы заполняются из системных представлений Oracle. При RECREATE_SCHEMA=true пересоздаются и наполняются заново.

Таблица	Назначение
TRANSPILER_OBJECTS	Список PL/SQL-объектов для обработки (FUNCTION, PROCEDURE, PACKAGE, PACKAGE BODY, TYPE, TYPE BODY). Заполняется из transpiler_root_obj.
TRANSPILER_SOURCE	Исходный код объектов, необходимых для обработки.
TRANSPILER_ALL_DEPENDENCIES	Снапшот из dba_dependencies.
TRANSPILER_DEPENDENCIES	Зависимости между объектами. Заполняется из TRANSPILER_ALL_DEPENDENCIES с учётом ограничений TRANSPILER_ROOT_OBJ и TRANSPILER_STOP_OBJ.
TRANSPILER_SYNONYMS	Сведения о синонимах среди зависимостей.
TRANSPILER_TAB_COLUMNS	Описание колонок таблиц и представлений, используемых обрабатываемыми объектами.

Источники данных: DBA_OBJECTS, DBA_SOURCE, DBA_DEPENDENCIES, DBA_SYNONYMS, DBA_TAB_COLUMNS.

6. Вспомогательные таблицы

Необходимо создать и заполнить перед запуском. Скрипты создания будут добавлены в поставку отдельным SQL-файлом.

TRANSPILER_ROOT_OBJ – Задаёт входные точки - корневые объекты, от которых начинается обход зависимостей.

```
create table transpiler_root_obj (
  owner VARCHAR2(128 BYTE) not null,
```

```

name  VARCHAR2(128 BYTE) not null,
type  VARCHAR2(18)        not null, -- тип объекта или '%'
match CHAR(1) default 'E'      -- 'E' exact, 'L' like
);
alter table transpiler_root_obj
add constraint transpiler_root_obj_pk
primary key (owner, name, type, match);

```

Значения колонки match:

- 'E' (exact) — точное соответствие имени
- 'L' (like) — маска в синтаксисе LIKE, например SCS_CO%

Вместо схемы и типа можно использовать '%'. Если таблица пустая — результат сбора зависимостей тоже будет пустым.

Пример:

```

insert into transpiler_root_obj (owner, name, type, match)
values ('TBG', 'SCS_CO%', 'PACKAGE', 'L');

```

TRANSPILER_STOP_OBJ – Задаёт стоп-объекты. Построение зависимостей для них, включая все их последующие зависимости, ограничивается, то есть не добавляются тела пакетов и тела объектных типов.

Пример: А вызывает В, В вызывает С, С вызывает D, В — стоп-объект.

А, В, D – пакеты. С – процедура. Тогда построится цепочка зависимостей:

А-спецификация_пакета -> А-тело_пакета

А-спецификация_пакета -> В-спецификация_пакета -> С -> D-спецификация_пакета

```

create table transpiler_stop_obj (
  owner  VARCHAR2(128 BYTE) not null,
  name   VARCHAR2(128 BYTE) not null,
  type   VARCHAR2(18)        not null,
  match  CHAR(1) default 'E'
);
alter table transpiler_stop_obj
add constraint transpiler_stop_obj_stop_obj_pk
primary key (owner, name, type, match);

```

Пример:

```

insert into transpiler_stop_obj (owner, name, type, match)
values ('TBG', 'MGC%', '%', 'L');

```

7. Первый запуск

- Запускать из командной строки (cmd)
- Создать схему для размещения технических таблиц (пункт 5)
- Создать таблицы TRANSPILER_ROOT_OBJ и TRANSPILER_STOP_OBJ
- Заполнить их для определения входных точек и стоп-объектов
- Запустить в режиме plsql-feature-check с RECREATE_SCHEMA=true
- Запустить в режиме java (RECREATE_SCHEMA=true — опционально)

8. Параметры запуска

Параметры передаются через переменные окружения или аргументы NAME=value. Аргументы командной строки приоритетнее.

9.1. Основные параметры

Параметр	Описание
DB_HOST	Хост Oracle
DB_PORT	Порт Oracle
DB_SERVICE	Сервис Oracle
TBG_USERNAME	Пользователь Oracle
TBG_PASSWORD	Пароль пользователя Oracle
BASE_DIRECTORY	Путь к каталогу результата (относительный или абсолютный)
DEFAULT_JAVA_PACKAGE	Java-пакет для создаваемых классов (например, com.example)
CURRENT_SCHEMA	Схема для хранения технических метаданных
SCHEMA_TO_PROCESS	Схема, объекты которой анализируются или конвертируются
OBJECTS_TO_PROCESS	Список конкретных объектов для обработки (через запятую), опционально
WITH_DEPENDENCIES	Обрабатывать ли зависимости OBJECTS_TO_PROCESS (true / false)
RECREATE_SCHEMA	Пересоздавать технические таблицы (true / false). По умолчанию false
TRANSLATION_MODE	Режим: java, plsql-feature-check, validate-sql
FORK_JOIN_POOL_THREADS	Число потоков параллельной обработки. По умолчанию 4

9.2. Дополнительные параметры для validate-sql

Параметр	Описание
IGNITE_HOST	Хост Platform V Datagrid
IGNITE_PORT	Порт Platform V Datagrid
IGNITE_USERNAME	Пользователь Platform V Datagrid
IGNITE_PASSWORD	Пароль Platform V Datagrid
IGNITE_SCHEMA	Схема Platform V Datagrid

9. Когда использовать RECREATE_SCHEMA=true

- При первом запуске
- После изменений в коде утилиты, влияющих на структуру технических таблиц

- После изменений в анализируемой схеме SCHEMA_TO_PROCESS

Если данные актуальны — используйте RECREATE_SCHEMA=false для ускорения повторных запусков. Это значение по умолчанию.

10. Рекомендуемый способ запуска

Рекомендуется использовать именованные параметры вида NAME=value. Общий вид:

```
plsql-transpiler-tool.(bat|sh) <mode>
DB_HOST=...          DB_PORT=...          DB_SERVICE=...
TBG_USERNAME=...     TBG_PASSWORD=...
CURRENT_SCHEMA=...   SCHEMA_TO_PROCESS=...
BASE_DIRECTORY=...
```

11. Примеры запуска

11.1. Режим java

```
plsql-transpiler-tool.(bat|sh) java ^
DB_HOST=ora-db-host ^
DB_PORT=1521 ^
DB_SERVICE=ORCLCDB ^
TBG_USERNAME=TRANSPILER ^
TBG_PASSWORD=TRANSPILER ^
CURRENT_SCHEMA=TRANSPILER ^
SCHEMA_TO_PROCESS=SWIP ^
BASE_DIRECTORY=./ ^
DEFAULT_JAVA_PACKAGE=com.example ^
RECREATE_SCHEMA=true ^
FORK_JOIN_POOL_THREADS=4
```

Результат: В BASE_DIRECTORY создаются каталоги по DEFAULT_JAVA_PACKAGE с сгенерированным Java-кодом.

С обработкой только выбранных объектов:

```
plsql-transpiler-tool.(bat|sh) java ^
DB_HOST=ora-db-host ^
DB_PORT=1521 ^
DB_SERVICE=ORCLCDB ^
TBG_USERNAME=TRANSPILER ^
TBG_PASSWORD=TRANSPILER ^
CURRENT_SCHEMA=TRANSPILER ^
SCHEMA_TO_PROCESS=SWIP ^
BASE_DIRECTORY=./transpiler-out ^
DEFAULT_JAVA_PACKAGE=com.example ^
OBJECTS_TO_PROCESS=PKG_ONE,PKG_TWO ^
WITH_DEPENDENCIES=true ^
RECREATE_SCHEMA=false ^
FORK_JOIN_POOL_THREADS=4
```

11.2. Режим plsql-feature-check

```
plsql-transpiler-tool.(bat|sh) plsql-feature-check ^
```

```
DB_HOST=ora-db-host ^
DB_PORT=1521 ^
DB_SERVICE=ORCLCDB ^
TBG_USERNAME=TRANSPILER ^
TBG_PASSWORD=TRANSPILER ^
CURRENT_SCHEMA=TRANSPILER ^
SCHEMA_TO_PROCESS=SWIP ^
BASE_DIRECTORY=C:\work\feature-check ^
RECREATE_SCHEMA=false ^
FORK_JOIN_POOL_THREADS=4
```

Результат: Файлы и логи со списком неподдерживаемых конструкций. summary.txt и файл для каждого проблемного объекта. Используется для оценки готовности к транспилации.

11.3. Режим validate-sql

```
plsql-transpiler-tool.(bat|sh) validate-sql ^
DB_HOST=ora-db-host ^
DB_PORT=1521 ^
DB_SERVICE=ORCLCDB ^
TBG_USERNAME=TRANSPILER ^
TBG_PASSWORD=TRANSPILER ^
CURRENT_SCHEMA=TRANSPILER ^
SCHEMA_TO_PROCESS=SWIP ^
BASE_DIRECTORY=./validate-sql ^
IGNITE_HOST=ignite-host ^
IGNITE_PORT=10800 ^
IGNITE_USERNAME=ignite_user ^
IGNITE_PASSWORD=ignite_password ^
IGNITE_SCHEMA=PUBLIC ^
RECREATE_SCHEMA=false ^
FORK_JOIN_POOL_THREADS=4
```

Результат: Файлы с ошибками по отдельным SQL-запросам и summary.txt с агрегированной статистикой по типам ошибок.

12. Формат результатов режима validate-sql

12.1 Файлы ошибок

Для каждого объекта с ошибками создаётся отдельный .sql-файл, содержащий:

- путь до запроса внутри Oracle-объекта
- текст SQL-запроса
- текст ошибки Ignite

12.2 Сводный файл

В подкаталоге summary формируется summary.txt — сгруппированная статистика по классам ошибок и частоте их возникновения.

12.3 Структура набора данных по одной ошибке

1) Путь до запроса:

```
[Schema: TBG] -> [PackageSpecification: COR_DBMS_LOCK_UTL]
-> [PackageBody: COR_DBMS_LOCK_UTL]
-> [FunctionDefinition: Find_Lockid_By_Name]
-> [SELECT DLA.LOCKID ...]
```

2) Выполняемый запрос:

```
SELECT DLA.LOCKID FROM sys.DBMS_LOCK_ALLOCATED DLA WHERE DLA.NAME = '1';
```

3) Текст ошибки:

```
Failed to validate query. From line 2, column 6 to line 2, column 28:
Object 'DBMS_LOCK_ALLOCATED' not found within 'SYS'
```

12.4 Поиск исходного запроса

- По пути до запроса — используя схему, пакет, функцию и т.д.
- По первой строке запроса — в теле/спецификации пакета, функции

Примечание: Приведённая первая строка может отличаться числом пробелов от исходной. При поиске попробуйте вариант с одним пробелом.

13. Замечания

- Параметры можно задавать через переменные окружения, а в bat||sh-файл передавать только режим запуска.
- При запуске создаётся лог-файл в том же каталоге, где расположен jar-файл.

Модуль миграции Database Migration Utility

Database Migration Utility

1. Назначение

Экспорт и миграция таблиц и данных из Oracle в Platform V Datagrid. Один запуск обрабатывает одну целевую БД согласно переданной конфигурации. При записи в Platform V Datagrid данные автоматически реплицируются в YDB.

2. Подготовка

1. Скопируйте db-migration-utility-1.0.jar и сопроводительные файлы на локальную машину.
2. Настройте подключение к Oracle в db-config.properties (URL, логин, пароль).
3. Убедитесь, что в БД-источнике созданы служебные таблицы конфигурации:
 - Распакуйте архив O2I_patch.zip
 - Выполните runme.sql
 - Лог установки — в файле runme.lst

3. Конфигурация (БД-источник)

Перед запуском в Oracle должны быть созданы следующие служебные таблицы:

Таблица	Назначение
O2I_MIGRATION_CONFIG	Хранит коды конфигураций миграции.
O2I_TABS_FOR_MIGRATION	Список таблиц для переноса. Привязка к конфигурации — по полю migration_group.

4. Запуск

Команда запуска утилиты:

```
%JAVA_HOME%\bin\java.exe
--add-opens java.base/java.lang=ALL-UNNAMED
--add-opens java.base/sun.nio.ch=ALL-UNNAMED
--add-opens java.base/java.util=ALL-UNNAMED
--add-opens java.base/java.util.concurrent=ALL-UNNAMED
--add-opens java.base/java.nio=ALL-UNNAMED
--add-opens java.base/java.io=ALL-UNNAMED
-jar .\db-migration-utility-1.0.jar <CONFIG_CODE> <FORCE>
```

Параметры командной строки:

Параметр	Описание	Пример
<CONFIG_CODE>	Код конфигурации из O2I_MIGRATION_CONFIG	TEST
<FORCE>	true — мигрировать все таблицы. false — только	true

Параметр	Описание	Пример
	таблицы со статусом ERROR из прошлого запуска.	

5. Требования к окружению

- Java Runtime Environment (версия, соответствующая сборке)
- Права на чтение в Oracle и запись в целевые БД
- Сетевая доступность между узлом запуска и БД

Серверный рантайм – Data Access Framework (DAF)

1. Порядок запуска

Запуск выполняется штатными средствами Platform V Datagrid

<https://platformv.sbertech.ru/docs/public/IGN/17.8.0/common/documents/installation-guide/installation.html>

Успешность запуска отслеживается по логам Platform V Datagrid.

При успешном запуске должна быть запись в логе:

```
*** Cluster environment is ready
```

В случае отсутствия записи в логе об успешном запуске необходимо разобраться в причинах неуспешного запуска по логам.

2. Особенности запуска

Нельзя переводить состояние кластера Platform V Datagrid из INACTIVE в ACTIVE вручную (control.sh). Flexsoft JDP автоматически переводит кластер в состояние ACTIVE.