

Tech Tips for HMI Ovation

Оглавление

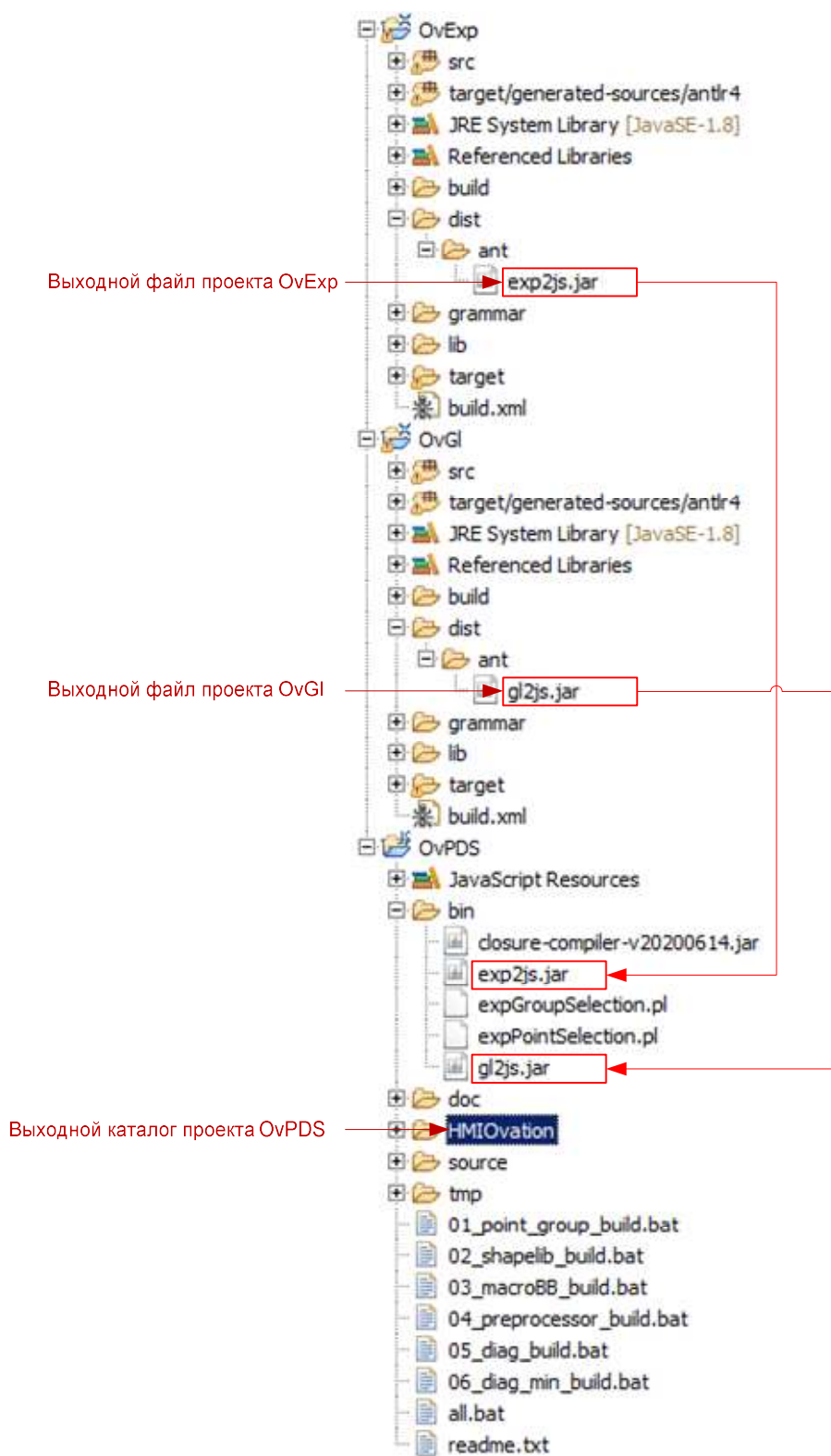
| | | |
|----|--|----|
| 1. | Файловая структура проекта HMIOvation | 2 |
| | Файловая структура Java проектов | 4 |
| | Файловая структура сценариев преобразования и визуализации графики Ovation | 5 |
| 2. | Сборка и установка Java проектов..... | 6 |
| | Сборка и установка exp2js.jar | 6 |
| | Сборка и установка gl2js.jar | 7 |
| 3. | Преобразование исходных данных Ovation..... | 8 |
| | Шаг 1 – подготовка данных точек и групп точек..... | 8 |
| | Шаг 2 – подготовка данных графической библиотеки форм..... | 9 |
| | Шаг 3 – подготовка данных о bounding box макросов | 10 |
| | Шаг 4 – интегрирование макросов в видеограммы | 11 |
| | Шаг 5 - трансформация видеограмм в формат js | 12 |
| | Шаг 6 – пакетирование и минифицирование видеограмм | 13 |
| 4. | Визуализации графики Ovation в web..... | 14 |

1. Файловая структура проекта NMIOvation

Проект NMIOvation состоит из следующих пакетов:

- Java проекты процессов преобразования исходных данных Ovation в формат JavaScript/JSON:
 - **OvExp** – процесс преобразования экспорт файла базы данных Ovation;
 - **OvGl** – процесс преобразования исходного кода графики Ovation;
- **OvPDS** - скрипты (сценарии):
 - преобразования исходных данных Ovation и
 - визуализации (рендеринга) графики Ovation в web.

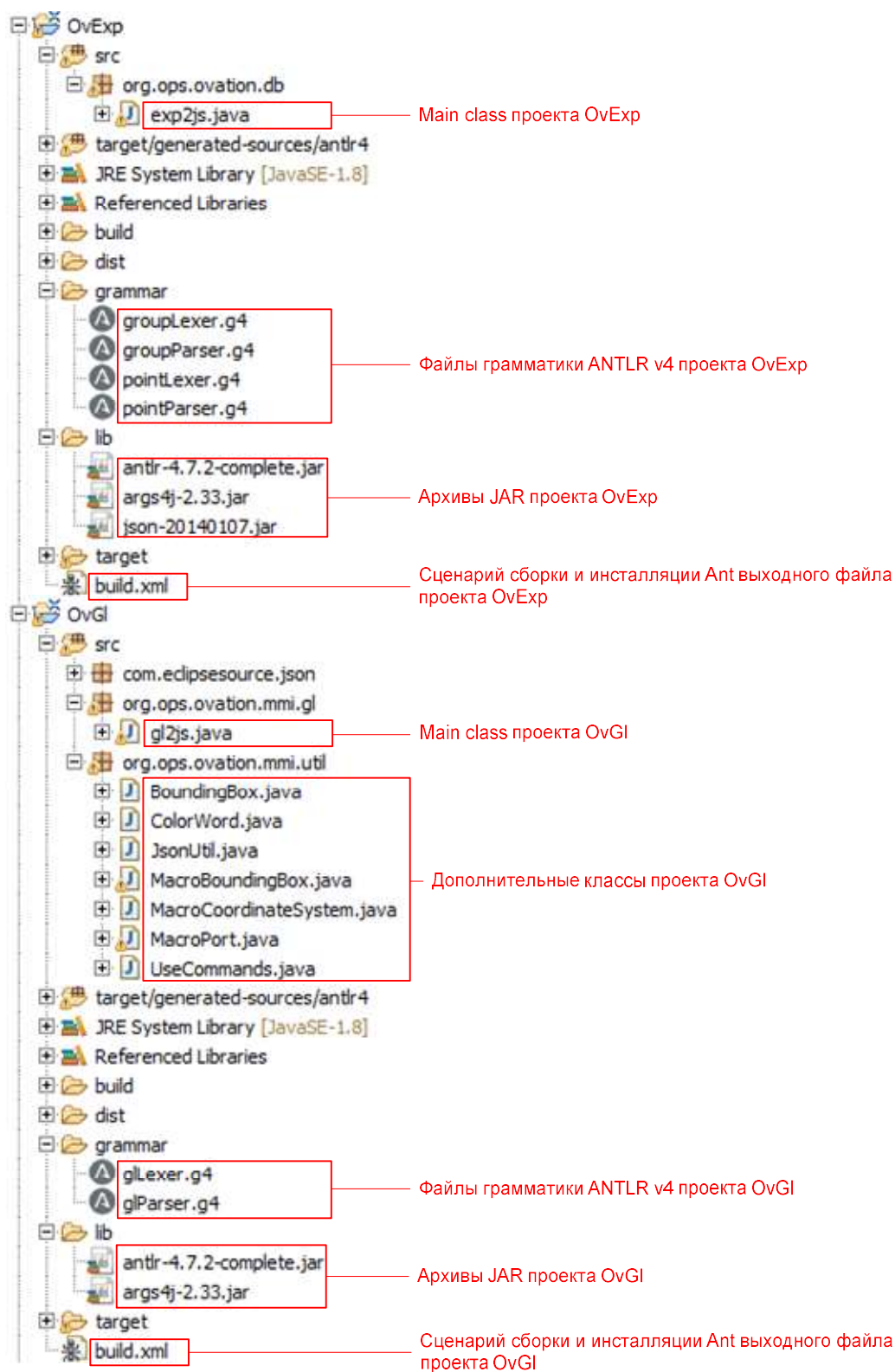
Файловая структура проекта NMIOvation представлена ниже.



Файловая структура проекта HMIOvation

Файловая структура Java проектов

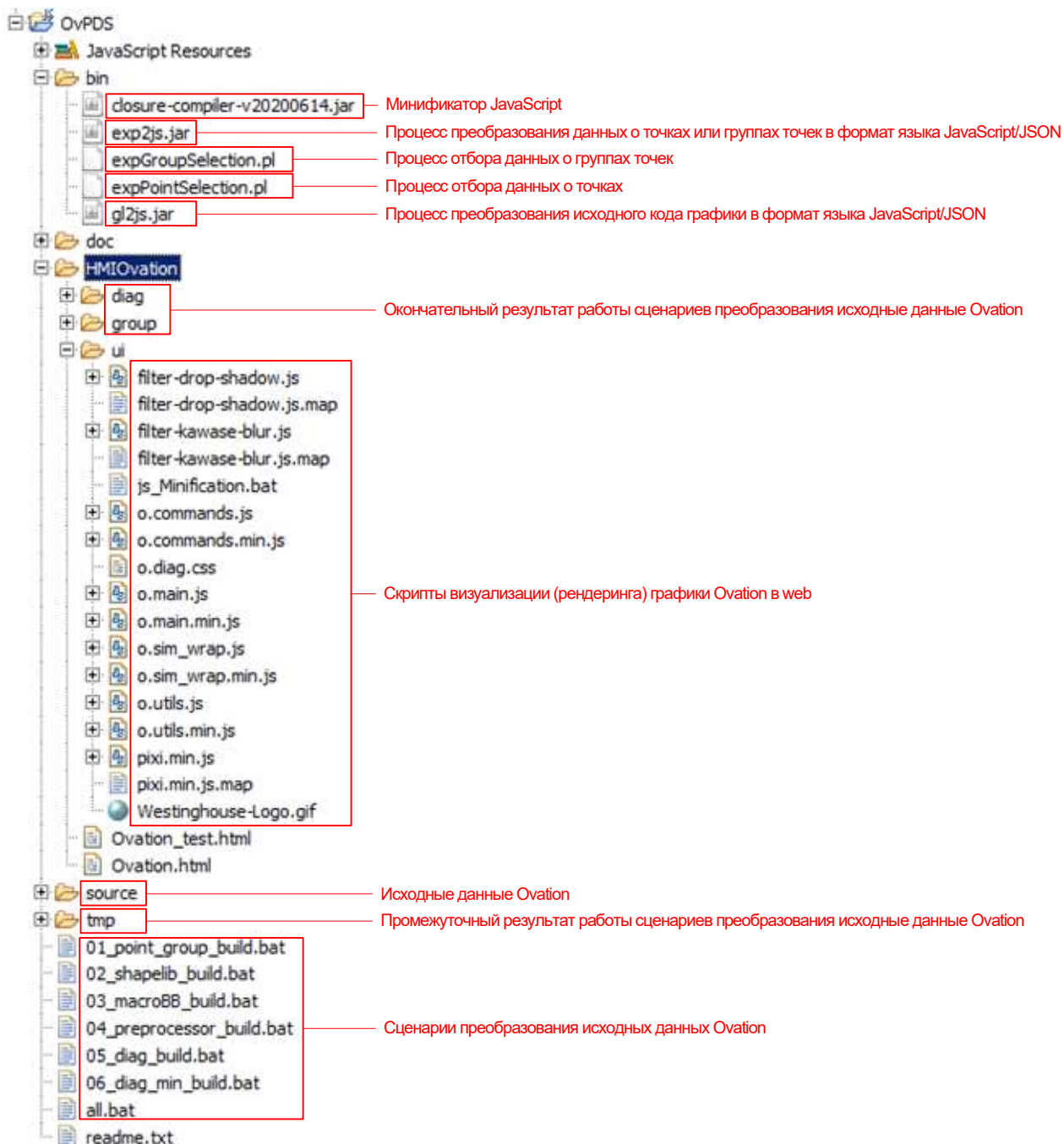
Файловая структура Java проектов представлена ниже.



Файловая структура Java проектов

Файловая структура сценариев преобразования и визуализации графики Ovation

Файловая структура сценариев представлена ниже.



Файловая структура сценариев

2. Сборка и установка Java проектов

Исполнительная среда сборки Java проектов:

- Windows / x64;
- Java Development Kit 8;
- Apache Ant 1.7.0 or higher.

Примеры сборки и установки Java проектов с использованием сборщика Ant представлены ниже.

Сборка и установка exp2js.jar

```
C:\Program Files\Java>ant -version
Apache Ant(TM) version 1.10.7 compiled on September 1 2019

D:\...\ts\OvExp>ant -f build.xml
Buildfile: D:\myProjects\OvExp\build.xml

clear:
  [echo] Delete ./build/classes and ./target/generated-sources/antlr4/org
  [delete] Deleting directory D:\myProjects\OvExp\build\classes
  [delete] Deleting directory D:\myProjects\OvExp\target\generated-sources\antlr4\org

translate:
  [echo] Translate ANTLR v4 ./grammar/*.g4 files into Java parser, lexer

initialize:
  [echo] Make directory ./build/classes
  [mkdir] Created dir: D:\myProjects\OvExp\build\classes

compile:
  [echo] Compiling source files to ./build/classes/
  [javac] Compiling 9 source files to D:\myProjects\OvExp\build\classes

create_run_jar:
  [echo] Create Runnable Jar for Project OvExp - ./build/ant/exp2js.jar
  [jar] Building jar: D:\myProjects\OvExp\dist\ant\exp2js.jar

BUILD SUCCESSFUL
Total time: 9 seconds

D:\...\ts\OvExp>ant -f build.xml install
Buildfile: D:\myProjects\OvExp\build.xml

install:
  [echo] Install ./dist/ant/exp2js.jar to ../OvPDS/bin/exp2js.jar
  [copy] Copying 1 file to D:\myProjects\OvPDS\bin

BUILD SUCCESSFUL
Total time: 0 seconds
```

Сборка и установка gl2js.jar

```
D:\...cts\OvG1>ant -f build.xml
Buildfile: D:\myProjects\OvG1\build.xml

clear:
  [echo] Delete ./build/classes and ./target/generated-sources/antlr4/org
  [delete] Deleting directory D:\myProjects\OvG1\build\classes
  [delete] Deleting directory D:\myProjects\OvG1\target\generated-sources\antlr4\org

translate:
  [echo] Translate ANTLR v4 ./grammar/*.g4 files into Java parser, lexer

initialize:
  [echo] Make directory ./build/classes
  [mkdir] Created dir: D:\myProjects\OvG1\build\classes
  [copy] Copying 1 file to D:\myProjects\OvG1\build\classes

compile:
  [echo] Compiling source files to ./build/classes/
  [javac] Compiling 27 source files to D:\myProjects\OvG1\build\classes

create_run_jar:
  [echo] Create Runnable Jar for Project OvG1 - ./build/ant/gl2js.jar
  [jar] Building jar: D:\myProjects\OvG1\dist\ant\gl2js.jar

BUILD SUCCESSFUL
Total time: 10 seconds

D:\...cts\OvG1>ant -f build.xml install
Buildfile: D:\myProjects\OvG1\build.xml

install:
  [echo] Install ./dist/ant/gl2js.jar to ../OvPDS/bin/gl2js.jar
  [copy] Copying 1 file to D:\myProjects\OvPDS\bin

BUILD SUCCESSFUL
Total time: 0 seconds
```

3. Преобразование исходных данных Ovation

Шаг 1 – подготовка данных точек и групп точек

Входы:

...OvPDS\source**partial.exp** - текстовый файл, содержащий текущие конфигурационные данные, извлеченные из главной базы данных Ovation.

Процессы:

...OvPDS\01_point_group_build.bat – сценарий выполнения процессов шага 1.

...OvPDS\bin\expPointSelection.pl – процесс отбора данных о точках Ovation (т.е. объектов типа: AnalogPoint; DigitalPoint; PackedPoint; DropPoint; AlgorithmPoint).

...OvPDS\bin\expGroupSelection.pl – процесс отбора данных о группах точек Ovation (т.е. объектов типа: PDSGroup).

...OvPDS\bin\exp2js.jar – процесс преобразования данных о точках или группах точек в формат языка JavaScript/JSON.

Выходы:

...OvPDS\tmp**PDSPoints.json** – текстовый JSON файл всех точек Ovation.

...OvPDS\HMIOvation\group**NNN.js** – текстовые JavaScript файлы групп точек Ovation.

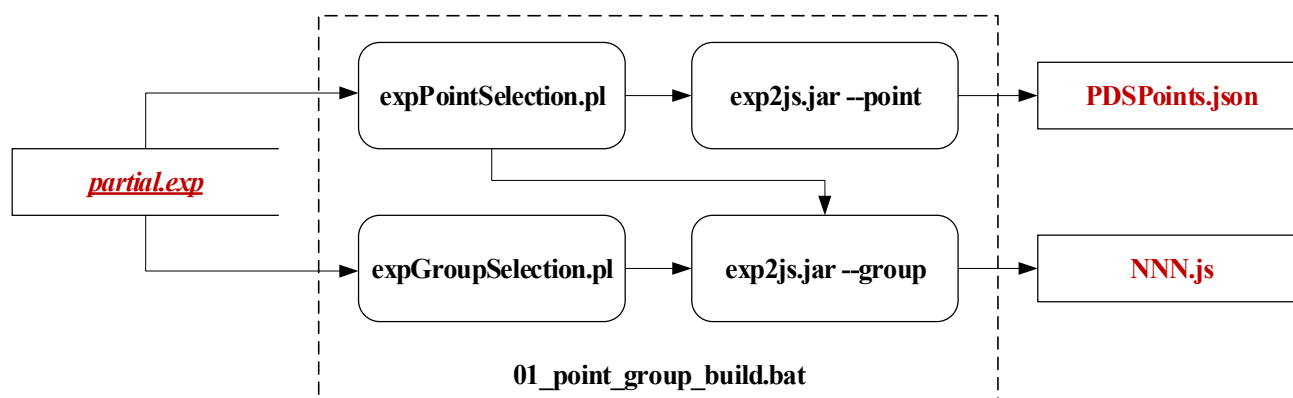


Диаграмма подготовка данных о точках и группах точек

Примечание: Время выполнения сценария ~ **30 сек.**

Нулевая длина файла ...OvPDS\01.log подтверждает правильность подготовки данных.

Шаг 2 – подготовка данных графической библиотеки форм

Входы:

...OvPDS\source**shapelib.src** – текстовый файл исходного кода графической библиотеки форм (**shapelib**), используемой HMI Ovation.

Процессы:

...OvPDS\02_**shapelib_build.bat** – сценарий выполнения процессов шага 2.

...OvPDS\bin**gl2js.jar** – процесс преобразования исходного кода графики Ovation в формат языка JavaScript/JSON.

Выходы:

...OvPDS\tmp**o.shapelib.js** – текстовый JavaScript файл графической библиотеки форм Ovation, предназначен для последующего включения JavaScript кода в файл ...OvPDS\HMIOvation\ui**o.utils.js**.

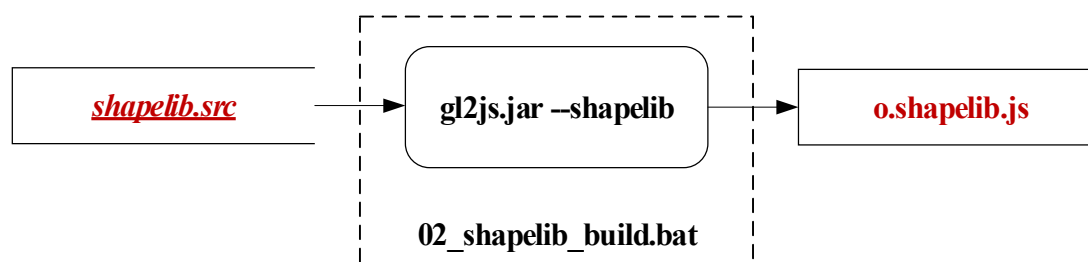


Диаграмма подготовка данных графической библиотеки форм

Примечание: Время выполнения сценария ~ **15 сек.**

Нулевая длина файла ...OvPDS\02.log подтверждает правильность подготовки данных.

Шаг 3 – подготовка данных о bounding box макросов

Входы:

...OvPDS\source**macro*.src** – текстовые файлы исходного кода графических макросов (**macro**), используемых на видеограммах подсистемы HMI Ovation.

Процессы:

...OvPDS\03_macroBB_build.bat – сценарий выполнения процессов шага 3.

...OvPDS\bin\gl2js.jar – процесс преобразования исходного кода графики Ovation в формат языка JavaScript/JSON.

Выходы:

...OvPDS\tmp**PDSMacro.json** – текстовый JSON файл габаритных размеров контейнеров графических макросов Ovation.

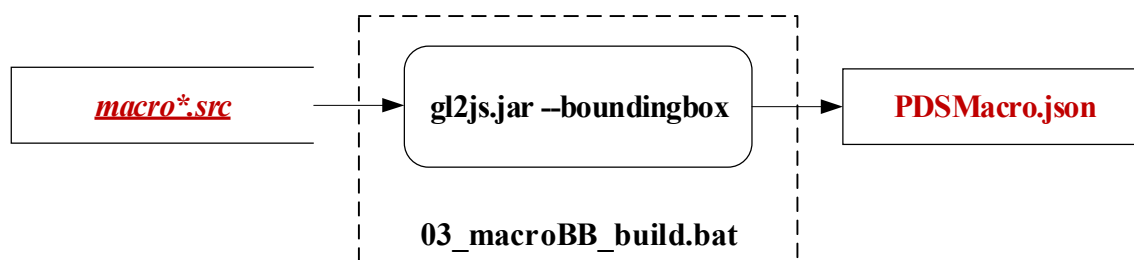


Диаграмма подготовка размеров контейнеров графических макросов

Примечание: Время выполнения сценария ~ **10 мин.**

Нулевая длина файла ...OvPDS\03.log подтверждает правильность подготовки данных.

Шаг 4 – интегрирование макросов в видеограммы

Входы:

...OvPDS\source**nnn[n].src** – текстовые файлы исходного кода графических видеограмм подсистемы HMI Ovation.

...OvPDS\source**macro*.src** – текстовые файлы исходного кода графических макросов (**macro**), используемых на видеограммах подсистемы HMI Ovation.

...OvPDS\tmp**PDSMacro.json** – текстовый JSON файл размеров контейнеров макросов (**macro**) HMI Ovation.

Процессы:

...OvPDS\04_preprocessor_build.bat – сценарий выполнения процессов шага 4.

...OvPDS\bin\gl2js.jar – процесс преобразования исходного кода графики Ovation в формат языка JavaScript/JSON.

Выходы:

...OvPDS\tmp**nnn[n].pp** – текстовые файлы исходного кода графических видеограмм с вложенными кодами макросов и их (макросов) габаритными размерами.

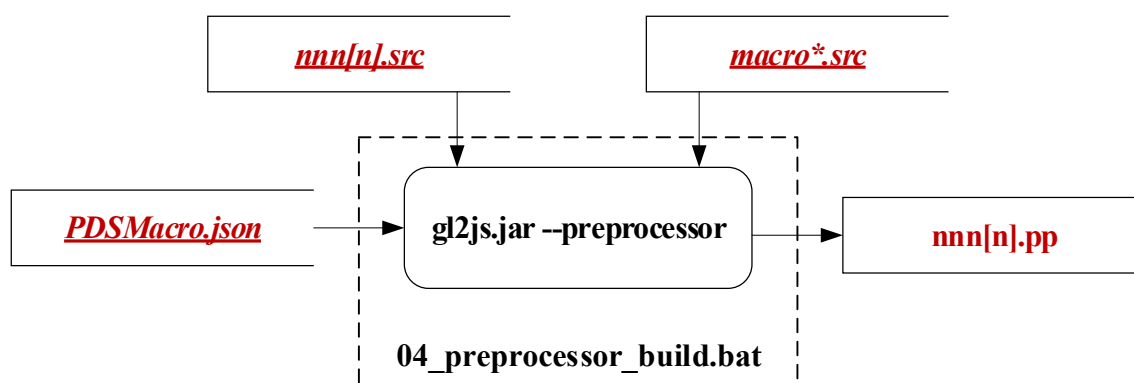


Диаграмма включения макросов в видеограммы

Примечание: Время выполнения сценария ~ **3 часа 45 мин.**

Нулевая длина файла ...OvPDS\04.log подтверждает правильность подготовки данных.

Шаг 5 - трансформация видеограмм в формат js

Входы:

...OvPDS\tmp**nnn[n].pp** – текстовые файлы исходного кода графических видеограм с вложенными кодами макросов и их (макросов) габаритными размерами.

...OvPDS\tmp**PDSPoints.json** – текстовый JSON файл всех точек Ovation.

Процессы:

...OvPDS\05_diag_build.bat – сценарий выполнения процессов шага 5.

...OvPDS\bin**gl2js.jar** – процесс преобразования исходного кода графики Ovation в формат языка JavaScript/JSON.

Выходы:

...OvPDS\tmp**nnn[n].js** – файлы видеограм в JavaScript формате.

...OvPDS\tmp**diag_bundler.cmd** – сценарий сборки JavaScript кодов главного и требуемых всплывающих окон в единый пакет **diag**, для всех видеограм проекта.

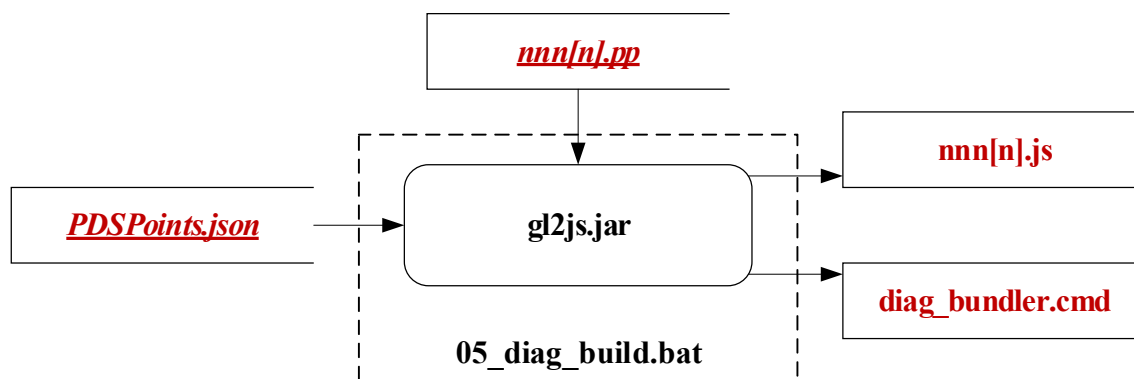


Диаграмма трансформации видеограмм в JavaScript код

Примечание: Время выполнения сценария ~ **1 час 10 мин.**

Нулевая длина файла ...OvPDS\05.log подтверждает правильность подготовки данных.

Шаг 6 – пакетирование и минифицирование видеограмм

Входы:

...OvPDS\tmp**nnn[n].js** – файлы видеограм в JavaScript формате.

Процессы:

...OvPDS\06_diag_min_build.bat – сценарий выполнения процессов шага 5.

...OvPDS\tmp**diag_bundler.cmd** – сценарий сборки JavaScript кодов главного и требуемых всплывающих окон в единый файл-пакет **diag**, для всех видеограм проекта.

...OvPDS\bin\closure-compiler-v20200614.jar – оптимизирующий компилятор JavaScript.

Выходы:

...OvPDS\tmp**nnn[n].diag** – файлы-пакеты видеограм **diag**.

...OvPDS\HMIOvation\diag**nnn[n].min.diag** – оптимизированные файлы-пакеты видеограм **diag**.

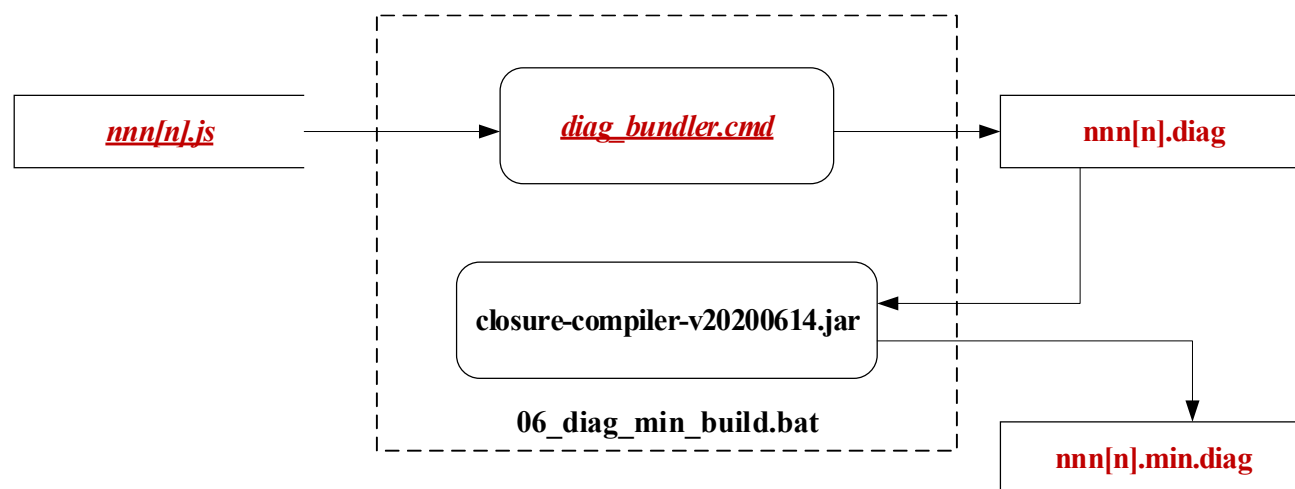


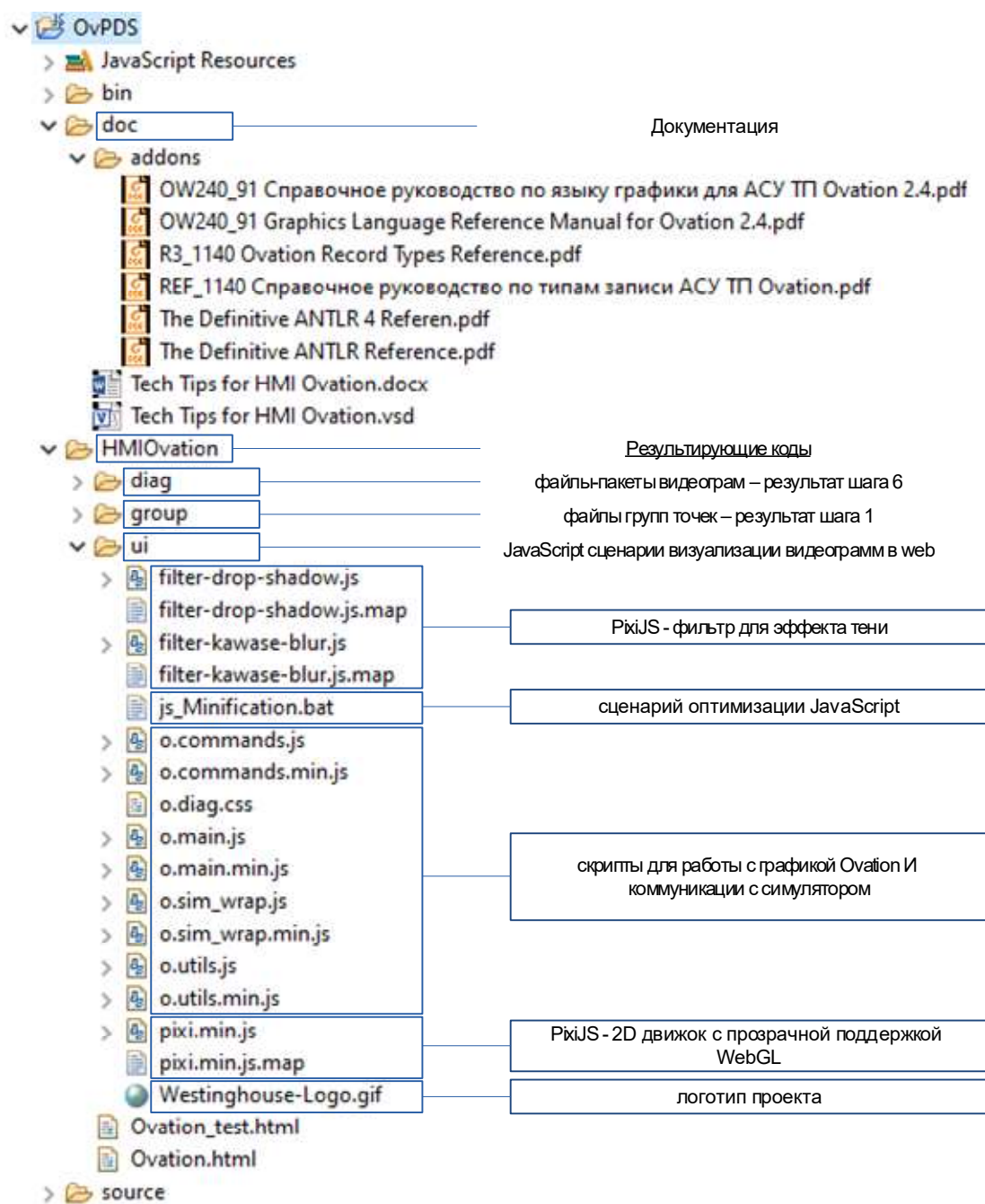
Диаграмма трансформации JavaScript кодов видеограмм в оптимизированный формат **diag**

Примечание: Время выполнения сценария ~ **20 мин.**

Нулевая длина файла ...OvPDS\06.log подтверждает правильность подготовки данных.

4. Визуализации графики Ovation в web

Файловая структура визуализации графики Ovation в web представлена ниже.



Олег Сергеев

sergoleg@gmail.com

sergeev@wsci-ua.com