# BPMN Конвертер: Что и как изменяется при преобразовании StormBPMN → Camunda

## Основные изменения

### 1. Добавляем Camunda namespaces

В корневой элемент <bpmn:definitions> добавляются:

xmlns:camunda="http://camunda.org/schema/1.0/bpmn"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

### 2. Обновляем процесс

В элемент <bpmn:process> добавляются атрибуты:

isExecutable="true"

camunda:historyTimeToLive="1"

### 3. Удаляем ненужные элементы

**Полностью удаляем:**

1. <bpmn:collaboration> - вся секция со всем содержимым
2. <bpmn:intermediateCatchEvent> - все промежуточные события
3. <bpmn:intermediateThrowEvent> - все промежуточные события
4. <bpmn:messageEventDefinition>
5. <bpmn:timerEventDefinition>
6. Связанные <bpmn:sequenceFlow> для удаленных элементов
7. Диаграммные элементы <bpmndi:BPMNShape> и <bpmndi:BPMNEdge> для удаленных объектов

**НЕ трогаем:**

1. <bpmn:endEvent> - конечные события остаются

### 4. Перенаправляем потоки

Когда удаляем промежуточное событие:

БЫЛО: TaskA → IntermediateEvent → TaskB

СТАЛО: TaskA → TaskB

**Алгоритм:**

1. Найти входящие потоки (где targetRef = ID события)
2. Найти исходящие потоки (где sourceRef = ID события)
3. Перенаправить входящие потоки на цели исходящих
4. Удалить все связанные потоки

### 5. Преобразуем задачи в serviceTask

**Все эти типы →** <bpmn:serviceTask>:

1. <bpmn:userTask>
2. <bpmn:manualTask>
3. <bpmn:callActivity>
4. <bpmn:businessRuleTask>
5. <bpmn:scriptTask>
6. <bpmn:sendTask>
7. <bpmn:receiveTask>
8. <bpmn:task>

**К каждой serviceTask добавляем:**

camunda:type="external"

camunda:topic="bitrix\_create\_task"

### 6. Добавляем условия к потокам

**Для потоков с name="да":**

<bpmn:conditionExpression xsi:type="bpmn:tFormalExpression">${result == "ok"}</bpmn:conditionExpression>

**Для потоков с name="нет":**

<bpmn:conditionExpression xsi:type="bpmn:tFormalExpression">${result != "ok"}</bpmn:conditionExpression>

### 7. Встраиваем ответственных (если есть JSON)

Если рядом с файлом лежит {имя\_файла}\_assignees.json, то в соответствующие serviceTask добавляем:

<bpmn:extensionElements>

<camunda:properties>

<camunda:property name="assigneeName" value="Имя ответственного" />

<camunda:property name="assigneeId" value="12345" />

</camunda:properties>

</bpmn:extensionElements>

Формат JSON-файла:

{

"assigneeEdgeId": 15700296,

"assigneeName": "Рук.отд. арх.сопр. и анализа проектов",

"assigneeId": 15298311,

"elementId": "Activity\_1597r5e",

"elementName": "Поставить задачу",

"assigneeType": "HUMAN",

"duration": 900

}

Если для элемента назначено несколько ответственных, то будет встроен только первый.

### 8. Исправляем конфликты default flow

**Проблема:** Если у <bpmn:exclusiveGateway> есть атрибут default="Flow\_ID", а у потока Flow\_ID есть условие - это ошибка в Camunda.

**Решение:** Убираем атрибут default у шлюза, если соответствующий поток имеет <bpmn:conditionExpression>.

### 9. Упорядочиваем элементы

Внутри каждого BPMN элемента ставим дочерние элементы в правильном порядке:

1. <extensionElements>
2. <ioSpecification>
3. <property>
4. Все <incoming>
5. Все <outgoing>
6. Остальные элементы

## Что НЕ изменяется

1. ID процессов и элементов
2. Названия процессов и задач
3. exporterVersion
4. Конечные события (endEvent)
5. Корректные default атрибуты (без конфликтов)

### 10. Чек-листы

Для передачи чек-листа из Шторм в Битрикс24 необходимо в Шторме для выбранного Activity в описании разместить текст, отвечающей следующему регулярному выражению:

Название: ^ЧЕКЛИСТ:\s\*(.+)$

Пункты: ^\s\*•\s\*(.+)$

Имя чек-листа:

* ^ — начало строки.
* ЧЕКЛИСТ: — буквально строка «ЧЕКЛИСТ:».
* \s\* — любое количество пробелов/табов после двоеточия (может быть и ноль символов).
* (.+) — любой текста, который будет преобразован в имя чек-листа в Битрикс24
* $ — конец строки

Пункты:

* ^ — начало строки.
* \s\* — любое количество пробелов или табов перед символом.
* • — сам символ «маркер списка» (bullet).
* \s\* — любое количество пробелов после этого символа.
* (.+) —весь оставшийся текст строки (минимум один символ).
* $ — конец строки.

## Результат

**Входной файл:** my\_process.bpmn **Выходной файл:** camunda\_my\_process.bpmn

Выходной файл готов для импорта в Camunda Modeler и развертывания в Camunda Platform 7.