



Слияние (дерево).

Компонент к выбранному узлу основного дерева присоединяет данные из другого дерева. Комбинация настроек компонента позволяет:

- включить весь корневой узел присоединяемого дерева как дочерний в выбранный узел основного;
- включить подчиненные элементы корневого узла присоединяемого дерева как дочерние в выбранный узел основного;
- добавить корневой контейнер присоединяемого дерева в выбранный массив основного дерева в качестве нового элемента массива;
- добавить элементы корневого массива присоединяемого дерева в выбранный массив основного дерева в качестве новых элементов.

Один из возможных кейсов применения компонента — объединение данных из разных источников/систем/веб-сервисов в единую иерархическую структуру, решая такие задачи как *обогащение данных* и/или *консолидация источников*.

Порты

Вход

-  **Основное дерево** — порт для подключения дерева, в которое добавляются данные. Обязательный.
-  **Присоединяемое дерево** — порт для подключения дерева, данные которого присоединяются к основному. Обязательный.

Выход

-  **Выходное дерево** — результат слияния деревьев согласно выбранным настройкам.

Мастер настройки

Настройка компонента включает два этапа: *Настройку слияния деревьев* и *Настройку соответствия между деревьями данных*.

Настройка слияния деревьев

На странице мастера слева отображается структура основного дерева, справа (опционально) — структура присоединяемого дерева. В структуре основного дерева **выбирается узел-приемник** элементов присоединяемого дерева. Логика присоединения новых узлов в основное дерево задается следующими настройками:

- **Включить узел в** — выбор из списка типа узла-приемника:
 - *контейнер* — предполагается, что узел-приемник основного дерева является контейнером и корневой узел присоединяемого дерева или его дочерние узлы включаются в него в качестве подчиненных.
 - *массив* — предполагается, что узел-приемник основного дерева является массивом, в который добавляются один или несколько новых элементов из присоединяемого дерева.
- **Пропустить корневой узел присоединяемого дерева** — опция доступна только для узла-приемника типа *контейнер*:
 - при включенной опции в результате слияния **узел-приемник получает дочерние элементы корневого узла** присоединяемого дерева в качестве новых подчиненных узлов.
 - при выключенной опции **узел-приемник получает сам корневой узел** присоединяемого дерева в качестве подчиненного.

Элементы управления интерфейсом:

-  **Фильтрация** — поле ввода значения для фильтрации списка узлов в основном дереве.
-  **Только подходящие узлы** — отображает только те узлы, которые могут быть использованы в качестве узла-приемника.
-  **Показать узлы присоединяемого дерева** — опциональное отображение структуры присоединяемого дерева.

Примечание: После смены типа узла приемника в параметре *Включить узел в*, выбранный ранее узел может не удовлетворять новым настройкам. В этом случае узел выделяется красным как недоступный для выбора в качестве узла-приемника.

Настройка соответствия между деревьями данных

На второй странице мастера при необходимости осуществляется настройка структуры выходного дерева, реализующая логику настройки выходного порта. На данном этапе структуру дерева, полученного в результате слияния, можно дополнительно трансформировать.

Особенности слияния в Контейнер и Массив

Примечание: В дереве данных под путём узла понимается его положение относительно корневого узла. Полный путь — это цепочка имён родительских контейнеров и имя самого узла. Полный путь однозначно определяет местоположение узла в дереве данных.

Слияние в контейнер

1. Узел-приемник *Контейнер* не может быть массивом и не должен содержаться в массиве.
2. Корневой узел присоединяемого дерева не может быть массивом при включенной опции *Пропустить корневой узел присоединяемого дерева*.
3. Если в результате слияния у узла-приемника образуются подчиненные узлы с одинаковыми именами и путями, то для новых узлов из присоединяемого дерева автоматически генерируются уникальные имена путём добавления суффикса (`_1` , `_2` и т.д.).

Примеры

Пример 1

Настройка узла:

- Режим: Слияние в контейнер (настройка *Включить узел в: контейнер*).
- Узел-приёмник в основном дереве: `user` .
- Корневой узел присоединяемого дерева: `permissions` .
- Опция *Пропустить корневой узел присоединяемого дерева*: выключена.

Входные данные:

```
*/ данные (основное дерево) /*

≡ user
├─(ab) tag:      "admin"
└─(0/1) blocked: true

*/ данные (присоединяемое дерево) /*

≡ permissions
├─(0/1) packagePublish: true
└─ ≡ fileStorage
    ├─(0/1) read:  true
    └─(0/1) write: false
```

Результат слияния:

```
*/ схема данных (результат) /*
```

```
≡ user
```

```
└─(ab) tag
```

```
└─(0/1) blocked
```

```
└─ ≡ permissions
```

```
    └─(0/1) packagePublish
```

```
    └─ ≡ fileStorage
```

```
        └─(0/1) read
```

```
        └─(0/1) write
```

```
*/ данные (результат) /*
```

```
≡ user
```

```
└─(ab) tag:      "admin"
```

```
└─(0/1) blocked: true
```

```
└─ ≡ permissions
```

```
    └─(0/1) packagePublish: true
```

```
    └─ ≡ fileStorage
```

```
        └─(0/1) read:  true
```

```
        └─(0/1) write: false
```

В этом примере корневой узел присоединяемого дерева `permissions` целиком добавляется в узел `user` основного дерева. Поскольку опция *Пропустить корневой узел присоединяемого дерева* выключена, узел-приёмник `user` получает новый дочерний контейнер `permissions` со всеми вложенными узлами.

Пример 2

Настройка узла:

- Режим: Слияние в контейнер (настройка *Включить узел в: контейнер*).
- Узел-приёмник в основном дереве: `user.permissions` .
- Корневой узел присоединяемого дерева: `permissions` .
- Опция *Пропустить корневой узел присоединяемого дерева*: включена.

Входные данные:

```
*/ данные (основное дерево) /*
```

```
≡ user
```

```
└─(ab) tag: "admin"
```

```
└─ ≡ permissions
```

```
    └─ ≡ fileStorage
```

```
        └─(0/1) read: true
```

```
        └─(12) maxSessions: 5
```

```
*/ данные (присоединяемое дерево) /*
```

```
≡ permissions
```

```
└─ ≡ fileStorage
```

```
└─(0/1) read: false
```

```
└─(12) maxSessions: 10
```

```
└─(ab) comment: "temporary"
```

Результат слияния:

```
*/ схема данных (результат) /*
```

```
≡ user
├─(ab) tag
└─ ≡ permissions
    ├─ ≡ fileStorage
    │   └─(0/1) read
    ├─ ≡ fileStorage_1
    │   └─(0/1) read
    ├─(12) maxSessions
    ├─(12) maxSessions_1
    └─(ab) comment
```

```
*/ данные (результат) /*
```

```
≡ user
├─(ab) tag: "admin"
└─ ≡ permissions
    ├─ ≡ fileStorage
    │   └─(0/1) read: true
    ├─ ≡ fileStorage_1
    │   └─(0/1) read: false
    ├─(12) maxSessions: 5
    ├─(12) maxSessions_1: 10
    └─(ab) comment: "temporary"
```

В данном примере в узел-приемник `user.permissions` основного дерева добавляются подчиненные узлы корневого узла-контейнера присоединяемого дерева, поскольку опция *Пропустить корневой узел присоединяемого дерева* включена. В результате возникает конфликт узлов с одинаковыми именами, который разрешается путем добавления суффиксов к именам узлов из присоединяемого дерева.

Слияние в массив

1. Узел-приемник должен быть массивом и контейнером.
2. Узел-приемник не должен содержаться в массиве.
3. Если корневой узел присоединяемого дерева является массивом, то все его элементы добавляются в массив-приемник.
4. Если корневой узел присоединяемого дерева является контейнером и не является массивом, то он становится новым элементом массива-приемника.
5. При слиянии в массив предполагается, что корневой узел присоединяемого дерева имеет такую же структуру, что и элемент массива в основном дереве. Если есть различия, то формируется

обобщённая структура элементов массива, которая включает узлы из основного и из присоединяемого дерева, т.е. выполняется объединение схем имеющихся (в основном дереве) и новых элементов массива-приемника.

Объединение схем элементов массива и разрешение конфликтов объединения

При объединении схем на каждом иерархическом уровне схемы элемента массива-приемника для каждого контейнера (в том числе и контейнера-массива) формируется общий набор дочерних узлов.

При формировании общего набора узлов элементы обновленного массива-приемника могут получить новые узлы. Заполнение данными этих узлов происходит следующим образом:

- отсутствующий ранее в отдельном элементе массива и добавленный в результате объединения схем узел простого типа заполняется значением `null`.
- отсутствующий ранее в отдельном элементе массива и добавленный в результате объединения схем дочерний контейнер и/или массив в итоговое дерево данных не включается.

Таким образом, узлы простых типов, попавшие в схему в результате объединения, фактически ведут себя как «обязательные узлы» (всегда присутствуют в данных, но могут иметь значение `null`), а контейнеры и массивы — как «необязательные» (присутствуют в итоговой схеме, но могут отсутствовать в данных итогового дерева).

В результате добавления новых узлов могут возникать конфликты с ранее имеющимися в схеме узлами:

Если в результате объединения схем одноимённые узлы с одинаковыми путями имеют разные типы данных (в том числе свойства Массив, Контейнер), то в результирующей схеме для каждого типа создаются отдельные узлы путём добавления суффикса к имени элемента.

Различие Меток, Видов данных и Назначения не является конфликтом. Если в объединяемых элементах массива есть неконфликтующие узлы, у которых совпадают путь, имя и тип данных, то в результирующей схеме Метка, Вид данных и Назначение итогового узла берутся из основного дерева.

Примеры

Пример 3

Настройка узла:

- Режим: Слияние в массив (настройка *Включить узел в:* массив).
- Узел-приёмник в основном дереве: массив `tasks` внутри `user`.
- Корневой узел присоединяемого дерева: `task`.
- Опция *Пропустить корневой узел присоединяемого дерева*: недоступна в этом режиме.

Входные данные:

```
*/ схема данных (основное дерево) /*
```

```
≡ user
├─(ab) login
└─[≡] tasks
    ├─(12) id
    └─(ab) title
```

```
*/ данные (основное дерево) /*
```

```
≡ user
├─(ab) login: "admin"
├─ ≡ tasks
|   ├─(12) id: 1
|   └─(ab) title: "Настроить права"
└─ ≡ tasks
    ├─(12) id: 2
    └─(ab) title: "Импорт пользователей"
```

```
*/ данные (присоединяемое дерево) /*
```

```
≡ task
├─(12) id: 3
├─(ab) title: "Резервное копирование"
├─(9.0) estimateHours: 8.0
└─ ≡ details
    └─(12) priority: 1
```

Результат слияния:

```
*/ схема данных (результат) /*
```

```
≡ user
```

```
└─(ab) login
```

```
└─[≡] tasks
```

```
    └─(12) id
```

```
    └─(ab) title
```

```
    └─(9.0) estimateHours
```

```
    └─ ≡ details
```

```
        └─(12) priority
```

```
*/ данные (результат) /*
```

```
≡ user
```

```
└─(ab) login: "admin"
```

```
└─ ≡ tasks
```

```
|   └─(12) id:          1
```

```
|   └─(ab) title:       "Настроить права"
```

```
|   └─(9.0) estimateHours: null
```

```
└─ ≡ tasks
```

```
|   └─(12) id:          2
```

```
|   └─(ab) title:       "Импорт пользователей"
```

```
|   └─(9.0) estimateHours: null
```

```
└─ ≡ tasks
```

```
    └─(12) id:          3
```

```
    └─(ab) title:       "Резервное копирование"
```

```
    └─(9.0) estimateHours: 8.0
```

```
    └─ ≡ details
```

```
        └─(12) priority: 1
```

При слиянии в массив `user.tasks` добавляется новый элемент, состоящий из содержимого узла `task` присоединяемого дерева. При этом схема элементов массива-приёмника объединяется со схемой нового элемента: появляются новое поле простого типа `estimateHours` и новый дочерний контейнер `details`. Для существующих задач `estimateHours` заполняется `null`, а для новой задачи — значением `8.0`. Контейнер `details` в результирующих данных присутствует только в добавленном из присоединяемого дерева элементе массива, демонстрируя свойство «необязательности» контейнеров.

Пример 4

Настройка узла:

- Режим: Слияние в массив (настройка *Включить узел в: массив*).

- Узел-приёмник в основном дереве: массив `tasks` внутри `user` .
- Корневой узел присоединяемого дерева: массив `tasks` .
- Опция *Пропустить корневой узел присоединяемого дерева*: недоступна в этом режиме.

Входные данные:

```

*/ схема данных (основное дерево) /*

≡ user
└─[≡] tasks
    └─(12) id
    └─(9.0) estimateHours

*/ данные (основное дерево) /*

≡ user
└─ ≡ tasks      // массив-приемник tasks имеет 1 элемент
    └─(12) id:      1
    └─(9.0) estimateHours: 5.0

*/ схема данных (присоединяемое дерево) /*

[≡] tasks
└─(12) id
└─(12) estimateHours

*/ данные (присоединяемое дерево) /*

└─ ≡ tasks      // 1-ый элемент массива tasks
|   └─(12) id:      2
|   └─(12) estimateHours: 8
└─ ≡ tasks      // 2-ой элемент массива tasks
    └─(12) id:      3
    └─(12) estimateHours: 12

```

Результат слияния:

```
*/ схема данных (результат) /*
```

```
≡ user
```

```
└─[≡] tasks
```

```
    └─(12) id
```

```
    └─(9.0) estimateHours
```

```
    └─(12) estimateHours_1
```

```
*/ данные (результат) /*
```

```
≡ user
```

```
└─ ≡ tasks
```

```
|   └─(12) id:          1
```

```
|   └─(9.0) estimateHours: 5.0
```

```
|   └─(ab) estimateHours_1: null
```

```
└─ ≡ tasks
```

```
|   └─(12) id:          2
```

```
|   └─(9.0) estimateHours: null
```

```
|   └─(12) estimateHours_1: 8
```

```
└─ ≡ tasks
```

```
    └─(12) id:          3
```

```
    └─(9.0) estimateHours: null
```

```
    └─(12) estimateHours_1: 12
```

В этом примере показан конфликт типов для узлов с одинаковым именем и путем при объединении схем элементов результирующего массива. В основном дереве поле `estimateHours` в элементах массива `user.tasks` имеет тип Вещественный, а в присоединяемом дереве — тип Целый. При слиянии в массив по пути `tasks.estimateHours` возникают значения разных типов, поэтому в результирующей схеме создаются два поля:

- `estimateHours` — для значений вещественного типа в основном дереве,
- `estimateHours_1` — для значений целого типа из присоединяемого дерева.

Для элементов, у которых нет значения соответствующего типа, поля заполняются `null`. В примере из присоединяемого дерева добавляются два новых элемента массива `tasks`, и их значения `8` и `12` попадают в поле `estimateHours_1`, тогда как исходное значение `5.0` остаётся в поле `estimateHours` первого элемента.