

ООО «Прикладные системы»

Руководство настройщика по работе с программным
обеспечением

«Модульная автоматизированная система учета и контроля
«Atomic Keeper» на базе универсальной учетной платформы»

2024

СОДЕРЖАНИЕ

Условные обозначения и сокращения	3
1. Общие положения.....	4
2. Работа настройщика в ПО	6
3. Конфигурирование журналов.....	23
4. Конфигурирование отчетов	25
5. Действия при аварийных ситуациях.....	28
Приложение 1	30
Приложение 2	31
Приложение 3	47
Приложение 4	54
Приложение 5	56

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Сокращение (обозначение)	Расшифровка (пояснение)
АСУиК	Автоматизированная система учета и контроля
АЭС	Атомная электростанция
ИИИ	Источники ионизирующего излучения
ПО	Программное обеспечение
РАО	Радиоактивные отходы
СУБД	Система управления базами данных
УЕ	Учетная единица
УУП	Универсальная учетная платформа

1. ОБЩИЕ ПОЛОЖЕНИЯ

Руководство настройщика по работе с программным обеспечением «Модульная АСУиК «Atomic Keeper» на базе УУП» (далее – Руководство) содержит пошаговые инструкции и пояснения по основным операциям, выполняемым пользователем с ролью «Настройщик» в программном обеспечении «Модульная АСУиК «Atomic Keeper» на базе УУП» (далее – ПО). Основные возможности для пользователя с ролью «Настройщик» описаны в Разделе 2 настоящего документа.

1.1. Область применения

Программное обеспечение «Модульная АСУиК «Atomic Keeper» на базе УУП» (далее – ПО) предназначена для автоматизации процедур по учету и контролю источников ионизирующего излучения и радиоактивных отходов. ПО спроектировано как многопользовательское программное обеспечение на базе универсальной учетной платформы (далее – УУП) с соответствующей конфигурацией для учета источников ионизирующего излучения (далее – ИИИ), а также радиоактивных отходов (далее – РАО).

Областью применения программного обеспечения «Модульная АСУиК «Atomic Keeper» на базе УУП» являются организации, которые в своей деятельности связаны с обращением ИИИ и РАО.

1.2. Основные возможности ПО

ПО предоставляет следующие основные возможности:

Основная задача ПО – это учет, контроль и отчетность по ИИИ и РАО, данные о которых вводятся пользователем в операциях на основании первичных документов.

Основные функциональные возможности ПО:

1) Учет и регистрация – обработка и хранение информации о свойствах, характеристиках и выполняемых операциях по обращению с ИИИ и РАО, без возможности удаления.

2) Контроль – отслеживание за перемещением ИИИ, а также РАО от места их образования до места их окончательной обработки или хранения.

3) Мониторинг – предоставление информации в графическом виде о РАО, а также вспомогательном оборудовании, с возможностью интерактивного выбора любой учётной единицы РАО с детальным указанием ее свойств, истории изменений и др. информации.

4) Отчетность – генерирование отчетных документов по ИИИ и РАО для подачи в национальный регулятор, а также для внутреннего аудита и анализа.

5) Планирование и нормирование – формирование планов по переработке и нормированию образования РАО.

6) Поддержка работы с устройствами индикации вмешательства (УИВ) – ведение учета применяемых УИВ с интеграцией в операции, а также валидация выполнения операций с учетными единицами, на которые установлены УИВ.

7) Интеграция и обмен данными для автоматизации ручных действий и минимизирования ошибок при учете РАО. ПО поддерживает механизмы обмена данных со следующими внешними системами и устройствами (но не ограничиваясь):

- считыватели штрихкодов/QR-кодов;
- загрузка файлов из системы паспортизации РАО;
- каталог пользователей по протоколу LDAP/openLDAP.

1.3. Уровень подготовки пользователя.

Пользователь обязан знать:

- настоящее Руководство;
- навыки работы с ОС семейства Microsoft Windows и Microsoft Office;
- соответствующую терминологию настоящего документа.

1.4. Предварительные условия для работы с ПО.

Перед началом работы с ПО администратором установлено соответствующее программное обеспечение «Модульная АСУиК «Atomic Keeper» на базе УУП», входящее в поставку.

Убедиться, что на автоматизированном рабочем месте пользователя установлен браузер Google Chrome (версии 105 и выше) и пользователь имеет доступ к ПО в соответствии с его полномочиями.

В ПО зарегистрированы необходимые пользователи согласно «Руководству администратора по работе с программным обеспечением «Модульная АСУиК «Atomic Keeper» на базе УУП».

2. РАБОТА НАСТРОЙЩИКА В ПО

Главная задача пользователя с ролью «Настройщик» является описание учетных единиц в соответствии с технологическим процессом УЕ и другими особенностями учета УЕ в организации. Описание конфигурации типов учетных единиц и их свойств выполняется в специальном конфигураторе.

Интерфейс конфигуратора делится на две части:

- Левая часть содержит иерархическое дерево, с которыми будет работать пользователь с ролью «Настройщик».
- Правая часть конфигуратора содержит интерфейс, в котором указываются параметры и добавляются поля, описывающие свойства типов учетных единиц из дерева.

Дерево конфигурации представляет собой иерархическую модель данных, основанную на типах учетных единиц. Первоначально, конфигурация содержит один единственный «Базовый тип», от которого начинается ветвление. В данный тип невозможно вносить изменения, добавлять категории и поля.

Новый тип добавляется в необходимую ветвь дерева, если добавленный тип является дочерним, он наследует свойства и категории от родителя.

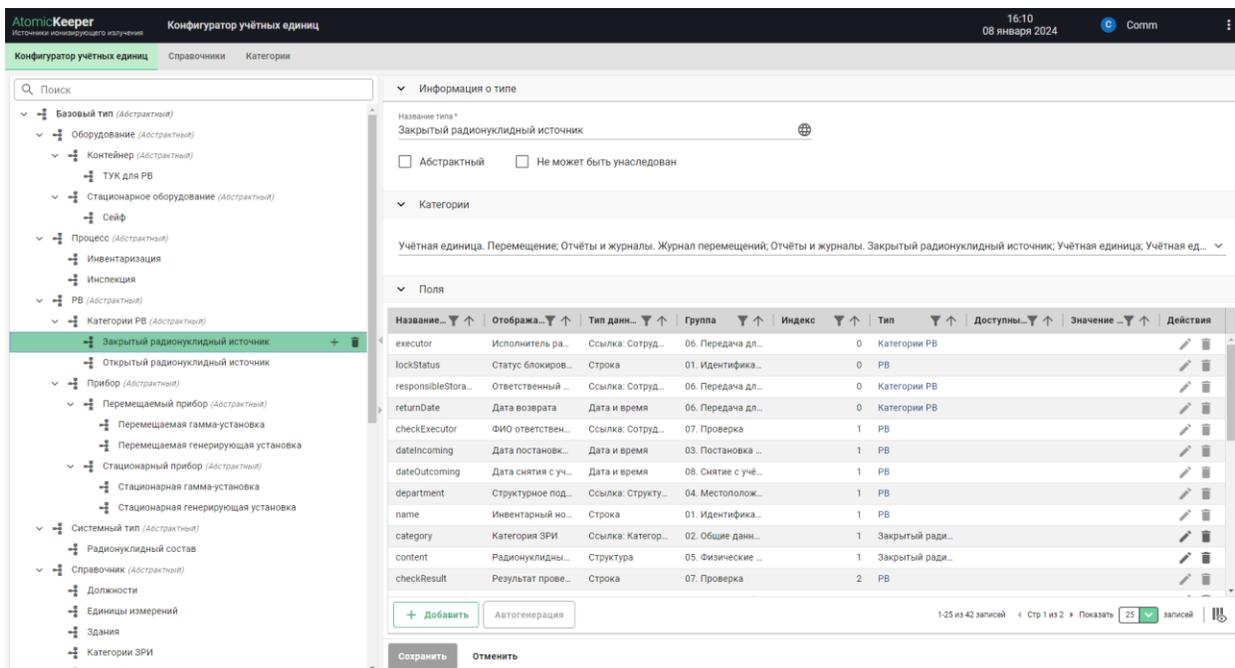


Рисунок 2.1. Пример интерфейса конфигурации по учету источников ионизирующего излучения

2.1. Работа с типами.

1. Добавление нового типа происходит нажатием кнопки **Добавить** на выделенном типе. После этого появляется новая строка. Далее необходимо ввести название типа в поле «Название типа». После ввода нового названия необходимо нажать кнопку **Сохранить** для добавления нового типа в дерево конфигурации.

2. Отмена добавления нового типа происходит нажатием на кнопку **Отменить**.

3. Для удаления существующего типа следует выделить необходимый тип и нажать кнопку **Удалить** ()

4. Подтвердить удаление с помощью кнопки **Сохранить** ()

Любые изменения, которые производит пользователь над типом можно сохранить, путем нажатия кнопки **Сохранить**, или отменить, при нажатии кнопки **Отменить**.

Каждый тип имеет дополнительные параметры, которые представлены в таблице и настраиваются по усмотрению пользователя:

Наименование параметра	Описание
Абстрактный	Параметр указывающий, что данный тип не может быть объектом для взаимодействия.
Не может быть унаследован	Параметр указывающий, что для данного типа нельзя создавать наследника.

2.2. Работа с категориями.

Категория – это требования, которые гарантируют наличие необходимых полей и их характеристик, которые должны быть у определённого типа. Также категория может активировать необходимую логику для типа учетной единицы.

Категории в ПО могут быть платформенными, либо «приносятся» модулями, которые обладают необходимой логикой для совершения действий в ПО.

Список доступных для работы категорий в ПО «Настройщик» может посмотреть на вкладке «Категории».

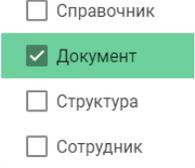
Название	Описание	Поля категории	Версия	Модуль
Учётная единица	Базовая учётная единица.	name, status, lockStatus, dateIncoming, dateOutcomi...	1	Учётная единица
Учётная единица. Контейнер	Учётная единица, в которую могут помещаться др...		1	Учётная единица
Учётная единица. Расходование	Учётная единица, количество материала в которо...		1	Учётная единица
Учётная единица. Инспекция	Учётная единица, которая используется для инсп...	dateIncoming, dateOutcoming, department, inspectio...	1	Учётная единица
Учётная единица. Инвентаризация	Учётная единица, которая используется для инве...	dateIncoming, dateOutcoming, department, inventoryI...	1	Учётная единица
Учётная единица. Перемещение	Учётная единица, которую можно переместить ил...	location, tempLocation	1	Учётная единица
Учётная единица. Проверка	Учётная единица, которую можно проверить	checkResult, nextCheckDate	1	Учётная единица
Отчёты и журналы. Категории ЗРИ	Категория для справочника "Категории ЗРИ".	name, status	1	Отчёты и журналы
Отчёты и журналы. Страны	Категория для справочника "Страны".	name, code	1	Отчёты и журналы
Отчёты и журналы. Структурные подразделения	Категория для справочника "Структурные подраз...	name, status	1	Отчёты и журналы
Отчёты и журналы. Приборы	Категория, которая определяет учётную единицу к...	name, location	1	Отчёты и журналы
Отчёты и журналы. Стационарное оборудование	Категория, которая определяет учётную единицу к...	name, location	1	Отчёты и журналы
Отчёты и журналы. Местонахождения УЕ	Категория, которая предоставляет учётным едини...	name, location	1	Отчёты и журналы
Отчёты и журналы. Изготовители	Категория для справочника "Изготовители".	name, country	1	Отчёты и журналы
Отчёты и журналы. Журнал перемещений	Учётные единицы с данной категорией отобража...	name, accountingNumber, type, department, dateInco...	1	Отчёты и журналы
Отчёты и журналы. Открытый радионуклидный ис...	Категория, которая определяет учётную единицу к...	name, fullName, type, department, factoryNumber, pa...	1	Отчёты и журналы
Отчёты и журналы. Получатели	Категория для справочника "Получатели".	name, country	1	Отчёты и журналы
Отчёты и журналы. Помещения	Категория для справочника "Помещения".	name, fullName, department, status	1	Отчёты и журналы
Отчёты и журналы. Закрытый радионуклидный ис...	Категория, которая определяет учётную единицу к...	name, type, department, factoryNumber, passportNu...	1	Отчёты и журналы
Отчёты и журналы. Поставщики	Категория для справочника "Поставщики".	name, country	1	Отчёты и журналы
Отчёты и журналы. Транспортные контейнеры	Категория, которая определяет учётную единицу к...	name, type, location	1	Отчёты и журналы
Отчёты и журналы. Типы ИИИ	Категория для справочника "Типы ИИИ".	name, status	1	Отчёты и журналы
Справочник	Платформенная категория, которая указывает, чт...	name, status	1	Универсальная учётная платформа
Документ	Категория, назначенная справочнику, хранящему ...	name, description, template, language, documentTyp...	1	Универсальная учётная платформа
Структура	Категория для создания специальных типов учёт...		1	Универсальная учётная платформа

Рисунок 2.2. Пример активных (доступных) категорий на вкладке «Категории»

Для присваивания (активации) категории для типа учетной единицы необходимо:

1. Создать новый тип (см. раздел 2.1 настоящего Руководства), либо выбрать необходимый существующий тип учетной единицы.

2. Развернуть строку  и нажать на кнопку  в открывшемся окне.

3. В выпадающем списке необходимо выбрать одно или несколько значений .

4. Для подтверждения действия необходимо нажать кнопку **Сохранить**.

Некоторые категории могут потребовать наличия у данного типа обязательных полей.

Их можно добавить вручную или воспользоваться кнопкой **Автогенерация**. После генерации необходимых полей, пользователю остаётся заполнить недостающие значения.

Более подробная информация про работу с полями описывающих тип учётной единицы, содержится в разделе 2.3 настоящего Руководства.

Несколько категорий могут требовать наличия поля с одним и тем же именем. Добавление одного поля с требуемым именем является необходимым и достаточным условием работы нескольких категорий.

Удаление (деактивация) категории с типа учетной единицы:

1. Выбрать нужный тип.
2. Развернуть список категорий.
3. В выпадающем списке снять галочки с ненужных категорий.
4. Для подтверждения действия необходимо нажать кнопку

Сохранить.

Базовая функциональность ПО поддерживает следующие платформенные категории:

- Справочник – платформенная категория, которая необходима для присвоения типу учетной единицы статуса «Справочник» и отображение его на вкладке «Справочники» в конфигураторе.

Необходимые поля для категории «Справочник»:

Название поля	Описание
name	Наименование записи в справочнике.
status	Учётный статус справочника.

- Сотрудник - платформенная категория, которая формирует список сотрудников для привязки к учетной записи пользователя.

Необходимые поля для категории «Сотрудник»:

Название поля	Описание
name	Фамилия и инициалы сотрудника
firstName	Имя сотрудника
lastName	Фамилия сотрудника
middleName	Отечество сотрудника
EncodersName	Фамилия и инициалы сотрудника на английском языке
position	Должность сотрудника

- Права на доступ к учётным единицам – платформенная категория, с помощью которой формируется список областей доступа для работы с учётными единицами. Область доступа назначается пользователю с ролью «Учетчик» для разделения полномочий. Каждая область доступа отображается в разделе «Группы прав» у администратора.

Необходимые поля для категории «Права на доступ к учётным единицам»:

Название поля	Описание
name	Наименование области доступа
firstName	Фильтрационное поле. В данном поле указывается название поля, по значениям которого необходимо фильтровать учётные единицы. Если поле, название которого указано в fieldName, отсутствует у учётной единицы, то по умолчанию такая учётная единица попадает в итоговую выборку значений. Поле скрыто по умолчанию в отображении справочников для Учётчика. Пример: значение fieldName = department, означает, что все учётные единицы, у которых существует поле department фильтруются по значению из поля name этой записи справочника.

- Документ прикрепляемый – платформенная категория, с помощью которой формируется список прикрепляемых типов документов на вкладке «Документы» при проведении операции.

Необходимые поля для категории «Документ прикрепляемый»:

Название поля	Описание
name	Наименование документа.

- Документ генерируемый – платформенная категория, с помощью которой формируется список генерируемых Системой типов документов на вкладке «Документы» при проведении операции.

Необходимые поля для категории «Документ генерируемый»:

Название поля	Описание
name	Наименование документа.
template	Шаблон (прикрепляемый файл).

- «Структура» является платформенной категорией, которая необходима для отображения информации в учетной единице в виде таблицы.

Каждое поле в типе учетной единицы с активной категорией «Структура», будет являться наименованием графы в таблице. Количество граф будет зависеть от количества созданных полей.

Например: тип учетной единицы с активной категорией «Структура» содержит следующие поля: Радионуклиды, Активность. Такой тип будет в интерфейсе пользователя с ролью «Учетчик» отображаться следующим в виде таблицы с графами: Радионуклиды, Активность.

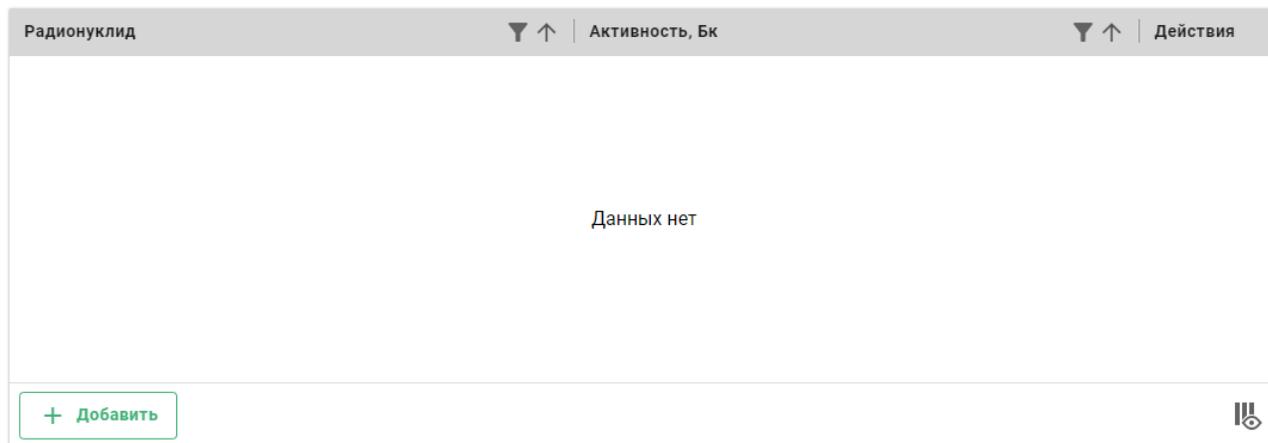


Рисунок 2.3. Пример отображения структуры у пользователя с ролью "Учетчик"

Необходимые поля для категории «Структура»:

Название поля	Описание
stuctId	Системное поле в виде счетчика, которое не заполняется и скрыто для пользователей. Данное поле необходимо для идентификации записей в таблице.

2.3. Работа с полями.

Поля представляют собой набор информации, который описывает свойства отдельного типа.

Каждому типу, кроме «Базового типа», могут быть добавлены поля.

Для этого необходимо:

1. Развернуть строку  Поля.
2. Нажать кнопку  в открывшемся окне.

3. В открывшемся окне «Добавление нового поля» заполнить данные.

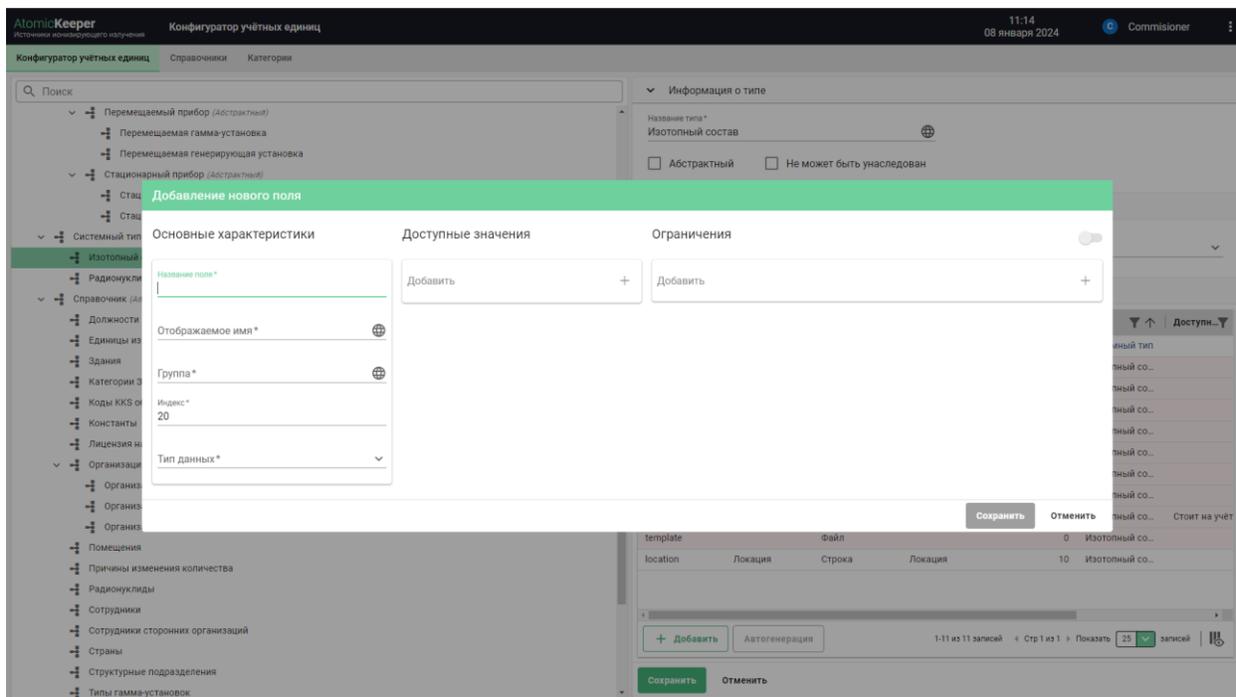
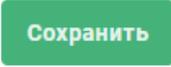


Рисунок 2.4. Интерфейс по добавлению нового поля

4. Нажать кнопку .

Для отмены создания нового поля необходимо нажать кнопку

.

В графе таблицы «Действия» расположены элементы управления (кнопки):

-  – редактировать;
-  – удалить.

При нажатии на кнопку **Удалить** поле окрашивается в красный цвет и появляются следующие элементы подтверждения в графе «Действия»:

-  – Сохранить;
-  – Отменить.

Окно «Добавление нового поля» содержит основной набор параметров, который приведён в таблице:

Наименование параметра	Описание
Название поля	<p>Системное название поля. Поле является обязательным для заполнения. Максимальное число символов — 64.</p> <p>Наименование поля должно быть уникальным.</p> <p>Разрешается в названии использовать только одно слово, состоящее из букв латинского алфавита, дефиса, цифры.</p>
Отображаемое имя	<p>Наименование, которое будет отражаться в интерфейсе пользователя. Поле является обязательным для заполнения.</p> <p>Максимальное число символов — 256.</p> <p>Разрешены буквы латинского и кириллического алфавита, цифры и спец.символы.</p>
Группа	<p>Группа, к которой принадлежит поле. Поле является обязательным для заполнения.</p> <p>Максимальное число символов — 64.</p> <p>Разрешены буквы латинского и кириллического алфавита, цифры и спец.символы.</p>
Тип данных	<p>Тип данных, который выбирается из выпадающего списка значений. Описание поддерживаемых типов данных выполнено в п.2.4.</p>
Множественные значения	<p>Указывает количество значений, которые может принимать поле и может принимать два состояния: Да/Нет.</p> <p>Нет – поле состоит из одного значения. При активном параметре «Допустимые значения» из списка возможно выбрать только одно значение.</p> <p>Да – одно поле может содержать одно или более значений. При активном параметре «Допустимые значения» из списка можно выбрать одно или несколько значений.</p> <p>Параметр не отображается, если выбран тип данных «Файл», «Обратная ссылка», «Логическая переменная».</p>

Поле уникальное	<p>Указывает, что значение в поле не может повторяться и является уникальным среди данного типа учётной единицы и его дочерних типов (при их наличии), которые наследуют данное поле, если уникальность определена в родительском типе.</p> <p>Уникальность проверяется вне зависимости от статуса учетных единиц.</p> <p>При проверке на уникальность учётная единица не сравнивается сама с собой.</p> <p>Параметр не отображается, если выбран тип данных «Файл», «Структура», «Обратная ссылка».</p>
Поле обязательное	<p>Указывает, что поле должно быть обязательным для заполнения.</p> <p>Отсутствует у типов данных: Логическая переменная, Обратная ссылка.</p>
Поле только для чтения	<p>Значения для данного поля можно внести однократно при проведении операции «Регистрация учётной единицы» или в операциях на ее основе.</p> <p>Далее, при работе с объектом, значение в данном поле будет использоваться только для чтения и не может быть изменено. Однако, пользователь может выделять его и копировать значение из него.</p> <p>Параметр не отображается, если выбран тип данных «Структура», «Обратная ссылка».</p> <p>Значения в типе поле с типом данных «Обратная ссылка заблокированы всегда»</p>
Поле заблокировано	<p>Указывает, что поле неактивно, пользователь не может взаимодействовать.</p> <p>Отсутствует у типов данных: Структура, Файл, Обратная ссылка.</p>
Автозаполняемое поле	<p>Параметр отображается, если выбран тип данных: Строка, Текст, Целое число, Дробное число, Дата и время, Дата.</p> <p>Указывает, что значение в поле может вычисляться Системой по заданной формуле на основе других полей.</p> <p>Добавление автоматического рассчитанного значения в поле происходит автоматически, если поле пустое либо</p>

	<p>после нажатия кнопки автозаполнение, которая размещена возле поля.</p> <p>Пользователь имеет возможность изменить значение, которое получилось в результате расчета, заменив расчетное значение пользовательским. При выполнении такого ввода кнопка автозаполнения становится неактивной.</p> <p>Список поддерживаемых функций и операторов указан в приложении 2</p>
Индекс	Указывает, какой приоритет отображения будет иметь добавляемое поле в пользовательском интерфейсе.
Скрыть поле в операциях и карточках	Поле, которое не отображается для заполнения и взаимодействия, а также отсутствует в карточке учётной единицы.

2.4. Типы данных.

«Тип данных» является обязательным для заполнения параметром. Данное ПО использует типы данных, которые приведены ниже:

Тип данных	Описание
Строка	Тип, при выборе которого доступен ввод любых символов.
Локализованная строка	Тип, при выборе которого доступен ввод любых символов и используется для отображения строки в зависимости от выбранного языка системы.
Текст	Тип, при выборе которого доступен ввод любых символов и используется для отображения многострочного текста.
Логическая переменная	Тип, при котором могут приниматься только два значения: «Да» или «Нет».
Целое число	Тип, при котором для ввода доступны только целочисленные значения.
Дробное число	<p>Тип, при котором для ввода доступны числа дробной часть. При этом отображаются только два знака после запятой.</p> <p>При вводе большого значения происходит округление и отображаемое значение принимает вид: 4.00e+5, что равно 400000.</p>

Дата и время	Тип, при котором для ввода доступны только дата и время.
Ссылка	Тип данных, при котором происходит ссылка на неабстрактный тип учетной единицы из дерева конфигурации.
Обратная ссылка	В поле с таким типом данных отображаются учётные единицы, которые имеют ссылки на данную учётную единицу. Данные в таком поле всегда заблокированные и представлены в виде ссылки на учетные единицы.
Таблица	Тип, при котором происходит ссылка на неабстрактный тип с категорией «Структура». Если этот параметр присвоить учётной единице, то при работе с ней данное поле будет отображаться в виде таблицы
Файл	Добавление в поле неисполняемых файлов размером не превышающим 50 МБ. Разрешённые форматы: <ul style="list-style-type: none"> • pdf; • docx; • xlxs; • тексты; • картинки.

В зависимости от значения, которое присвоит пользователь параметру «Тип данных», могут быть добавлены дополнительные параметры:

Дополнительный параметр	Тип данных	Описание
Доступные значения	<ul style="list-style-type: none"> • Строка • Локализуемая строка • Целое число • Дробное число • Дата и время 	Данный параметр даёт возможность задать список заранее определенных значений пользователю.
Значение по умолчанию	<ul style="list-style-type: none"> • Строка • Локализуемая строка 	Данный параметр даёт возможность задать значение по умолчанию. При активном параметр «Множественные

	<ul style="list-style-type: none"> • Логическая переменная • Целое число • Дробное число • Дата и время. 	значения» будет доступен выбор из введённых значений.
Значение для автозаполнения	<ul style="list-style-type: none"> • Строка • Текст • Целое число • Дробное число • Дата и время 	Поле для ввода формулы для вычисления значения, которая задается в виде valuePath. (См Приложение 1)
Название типа	<ul style="list-style-type: none"> • Ссылка • Обратная ссылка • Таблица 	Для типа данных Таблица в выпадающем списке отображаются все типы с категорией «Структура». Для типов данных «Ссылка», «Обратная ссылка» в выпадающем списке отображаются все неабстрактные типы УЕ не имеющие категорию «Структура». Выбранные значения используются как фильтр на какие типы может ссылаться поле либо какие типы попадают в поле с типом данных «Обратная ссылка»

В зависимости от значения, которое присвоит пользователь параметру «Тип данных», могут быть добавлены следующие ограничения на пользователем вводимые значения:

Оператор ограничения	Тип данных	Описание
=	<ul style="list-style-type: none"> • Строка • Текст • Целое число • Дробное число • Дата и время 	Значение поля должно быть равно заданному в ограничении значению, с учетом регистра для типов данных «Строка», «Текст».
≠	<ul style="list-style-type: none"> • Строка • Текст 	Значение поля должно быть не равно заданному в ограничении значению, с

	<ul style="list-style-type: none"> • Целое число • Дробное число • Дата и время 	учетом регистра для типов данных «Строка», «Текст».
<	<ul style="list-style-type: none"> • Целое число • Дробное число • Дата и время 	Значение поля должно быть меньше заданному в ограничении значению.
≤		Значение поля должно быть меньше либо равно заданному в ограничении значению.
>		Значение поля должно быть больше заданному в ограничении значению.
≥		Значение поля должно быть больше либо равно заданному в ограничении значению.
РЕГВЫРАЖ	<ul style="list-style-type: none"> • Строка • Текст 	Значение поля должно удовлетворять регулярному выражению заданному в ограничении.
ДЛСТР =	<ul style="list-style-type: none"> • Строка • Текст 	Количество символов в поле должно быть равно заданному в ограничении значению.
ДЛСТР ≠	<ul style="list-style-type: none"> • Строка • Текст 	Количество символов в поле должно быть неравно заданному в ограничении значению.
ДЛСТР <	<ul style="list-style-type: none"> • Строка • Текст 	Количество символов в поле должно быть меньше заданного в ограничении значения.
ДЛСТР ≤		Количество символов в поле должно быть меньше, либо равно заданному в ограничении значению.
ДЛСТР >		Количество символов в поле должно быть больше заданного в ограничении значения.
ДЛСТР ≥		Количество символов в поле должно быть больше, либо равно заданному в ограничении значению.

2.5. Добавление справочников.

Справочники выполняют две основные функции:

- они хранят и отображают нормативно-справочную информацию (например, список типов закрытых радионуклидных источников);
- на основании записей справочников формируются списки предопределённых значений, которые могут использоваться в остальных разделах данного ПО. Это позволяет не только сократить объем повторно вводимых данных, но и уменьшает вероятность внесения ошибок со стороны пользователя ПО.

Для создания справочника необходимо:

1. Создать тип, наименование которого будет наименованием справочника (см. раздел 2.1 настоящего Руководства).

Чтобы справочник отображался на вкладке «Справочники», необходимо параметр «Абстрактный» сделать неактивным.

2. Присвоить созданному типу категорию «Справочник» (см. раздел 2.2 настоящего Руководства).

3. Нажать кнопку **Автогенерация**.

4. Заполнить недостающие данные в созданных полях и добавить другие поля при необходимости (см. раздел 2.2 настоящего Руководства).

Заполнение вкладки «Справочники»:

1. Перейти на вкладку «Справочники».

2. Нажать кнопку **+Добавить**.

3. Заполнить необходимые поля.

4. Нажать **Сохранить**.

Для отмены ввода данных необходимо нажать кнопку **Отменить**.

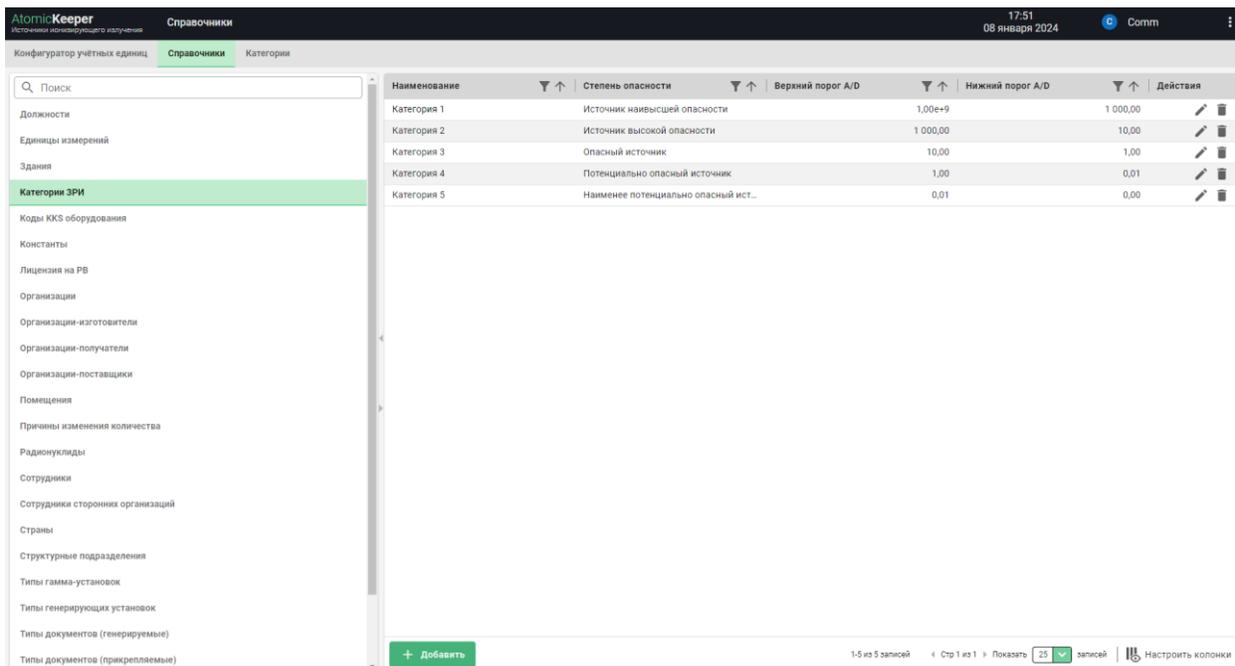


Рисунок 2.5. Пример созданных справочников в конфигурации по учету источников ионизирующего излучения

Редактирование вкладки справочников:

1. Перейти на вкладку «Справочники».
2. Выбрать справочник, в котором необходимо отредактировать объект.
3. Для редактируемого объекта в колонке «Действия» нажать кнопку

Редактировать ().

4. Отредактировать необходимые значения.

При редактировании справочников появляется дополнительный параметр Дата* 01.01.1970 × Время* 03:00 × . Параметр «Дата и Время» в справочниках отвечает за актуальность редактируемого объекта на определённый момент времени.

При ссылке на данный объект и активном параметре «Состояние объекта на момент времени» (см. раздел 2.4 настоящего Руководства), будет принято значение, которое было у объекта на момент времени, соответствующее параметру «Дата и Время».

Удаление объекта из справочника:

1. Перейти на вкладку «Справочники».
2. Выбрать справочник, в котором необходимо удалить объект.
3. Для удаляемого объекта в колонке «Действия» нажать кнопку **Удалить** ().

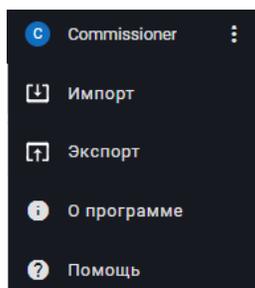
4. Для подтверждения действия нажать кнопку **Сохранить** ().
Для отмены удаления необходимо нажать кнопку **Отменить** ().

2.6. Экспорт конфигурации.

Экспорт конфигурации позволяет экспортировать дерево конфигурации и заполненные справочники.

Для экспорта конфигурации необходимо:

1. Войти в ПО пользователем с ролью «Настройщик».
2. Нажать на кнопку  в правом верхнем углу рядом с логином пользователя.



3. В выпавшем списке нажать кнопку **Экспорт**.
4. Экспортируется файл в формате ***.zip**.

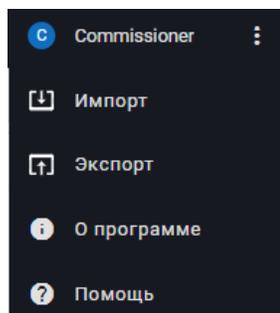
После выполнения данных действий появится загруженная конфигурация.

2.7. Импорт конфигурации.

Импорт конфигурации позволяет импортировать дерево конфигурации и заполненные справочники.

Для импорта конфигурации необходимо:

1. Войти в ПО пользователем с ролью «Настройщик».
2. Нажать на кнопку  в правом верхнем углу рядом с логином пользователя.



3. В выпавшем списке нажать кнопку **Импорт**.
4. Если в конфигураторе создавались типы, то появится валидационное сообщение рисунок 2.6.

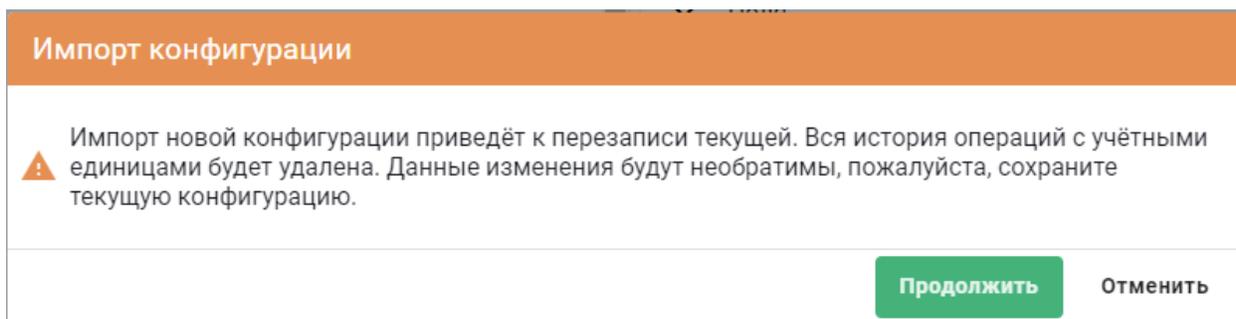


Рисунок 2.6. Валидационное сообщение при импорте конфигурации

5. Нажать **Продолжить**.
6. Выбрать файл в формате ***.zip**.

После выполнения данных действий появится экспортированная конфигурация.

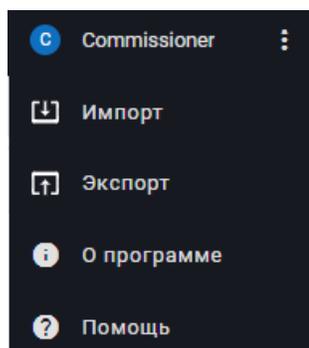
2.8. Раздел «Помощь».

Раздел «Помощь» даёт доступ к данному Руководству.

Для доступа к разделу «Помощь» необходимо:

1. Войти в ПО пользователем с ролью «Настройщик».

2. Нажать на кнопку  в правом верхнем углу рядом с логином пользователя.



3. В выпавшем списке нажать кнопку **Помощь**.

После выполнения данных действий откроется Руководство в формате ***.pdf**.

3. КОНФИГУРИРОВАНИЕ ЖУРНАЛОВ

Журналы используются для отражения изменений атрибутов, описывающих учётную единицу. Изменение атрибутов (полей) учётной единицы в Системе выполняется через операции. Формирование журнала происходит по изменению какого-либо одного или более триггерного поля, описывающего учётную единицу. Триггерное поле может быть сложным, если первоначальное триггерное поле является ссылкой. (Например: необходимо построить журнал перемещений ИИИ, который лежит в контейнере, а контейнер лежит в сейфе и сейф стоит в помещении. Для такой ситуации перемещение самого ИИИ, или перемещение контейнера с ИИИ, или перемещение сейфа должно инициировать запись в журнале перемещений ИИИ.)

Запись в журналах может отражать только новое состояние учётной единицы, предыдущее состояние отражается в предшествующей записи (-1 от новой) (Например, журнал учета ЗРИ, журнал учета ОРИ). Также запись в журнале может содержать информацию было-стало, т.е. в одной записи журнала есть информация и о новом состоянии учётной единицы и о предшествующем состоянии учётной единицы (Например журнал перемещений).

Настройка журнала в системе реализуется в Системе гибко. Журнал и его описание описывается в файле `customization.json`, который находится в папке с модулем «Отчеты и журналы» (По-умолчанию: `C:\Program Files\AppliedSystems\Atomic Keeper IRS\modules\reports-module`) и содержит следующие данные:

Таблица 3.1. Структура синтаксиса журнала

Параметр	Описание
<code>name</code>	Системное название журнала
<code>displayName</code>	Отображаемое название журнала. Пользовательское значение (перевод журнала на UI не происходит); Ключ для перевода (перевод ключа отображается согласно значениям из файла локализации)
<code>entityDefinitionIds</code>	ID типов учётных единиц, которые попадают в журнал
<code>triggerFields</code>	Триггерные поля учётных единиц, изменения которых создают запись в журнале. Структура описана в таблице П.3.1. приложения 3

fields	Поля, которые входят в журнал. Структура описания полей журнала зависит от значений в параметре source. Список поддерживаемых значений в source описан в таблице П.3.2., П.3.3. приложения 3.
postprocessing	Постпроцессинг - действия с журнал после его формирования. Как правило это фильтрация, сортировка и другие действия над данными журнала. Структура и поддерживаемые типы постпроцессинга описаны в таблице П.3.4. приложения 3.
permissions	Указывается набор прав для журнала. Поддерживаемый набор и структура их описания указан в таблице 3.2.

Таблица 3.2. Синтаксис прав для журнала

Параметр	Описание
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.view",	Право для просмотра журнала
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации) }.export"	Право для экспорта данных из журнала. При экспорте данных учитываются настройки фильтрации, сортировки, порядка и отображения колонок.

4. КОНФИГУРИРОВАНИЕ ОТЧЕТОВ

Настройка отчётов в системе в Системе делится на две взаимодополняемые части:

1. Описание отчета в json манифеста.

2. Создание и описание excel шаблона для отчета на основе данных, описанных в манифесте.

Первым делом настройщик в манифесте описывает логику и необходимые поля для отчёта, а затем в шаблоне excel указывает в нужных ячейках название описанных в манифесте полей, которые будут формировать данные для отчёта.

Манифест для отчётов содержат следующие данные:

- название отчета;
- id типов УЕ которые попадают в отчёт; (если не указан ни один id, в выборку не попадает ни один тип УЕ, т.е. отчёт отображается пустой)

- список полей
- список триггерных полей, если отчёт за период. (если триггерные поля не указаны для отчёта, который формируется за определенный период, то отчёт будет пустой);

- Логика сортировки и фильтрации записей (постпроцессинг).

Шаблон excel содержит следующие данные:

- Дефолтные данные:
 - названия отчета;
 - титульный лист для отчета с дефолтными значениями организации;

- шапка таблицы отчёта;
- другие дефолтные данные (утверждаю, согласовано и т.д.).
- Указать области данных в шаблоне в соответствии с приложением 4.
- Указание полей описанных в манифесте в необходимых ячейках, для формирования таблицы отчёта в виде `{{item["name поля из манифеста"]}}`;

- Необходимые теги ClosedXML (Приложение 5)

Процесс реализации логики работы (в системе)

- Найти все учётные единицы по типу, которые должны попасть в отчёт;

- Отфильтровать полученные УЕ:
 - если отчет на определенную дату, то в отчет приходят УЕ на дату отчёта;

- если отчет за период дат, то для УЕ, находятся изменения по всем триггерным полям;
- Отфильтровать и отсортировать УЕ по правилам, заданным в манифесте модуля (например: активность, лицензируемость);
- Выполнить логику сложных полей;
- Передать данные в шаблон excel и расположить их в необходимых ячейках согласно названиям полей в шаблоне.
- Сформировать отчет.

Таблица 4.1. Структура синтаксиса отчета:

Параметр	Возможные значения	Описание
isTemplateBased	false	Отчёт работает на новой механике, без подготовки сложного шаблона.
	true	Отчёт работает с помощью сложного шаблона закастомизированного через ClosedXML
name	пользовательское значение	Название отчёта с использованием пользовательского значения. Название отчета на UI не переводится при изменении локализации.
	ключ для перевода	Название отчёта с использованием ключа для перевода. Название отчета на UI переводится при изменении локализации. Перевод ключа отображается согласно значениям из файла локализации
type	пользовательское значение	Тип отчёта
templatae	{имя файла excel}. {ru}	Имя файла шаблона excel, который используется для формирования шаблона. После имени файла через точку указывается языковой префикс для определенной локализации. Т.е. при переключении языка на русский для формирования отчета будет использоваться шаблон {имя файла excel}. {ru}
entityDefinitionIds	ID типов УЕ	Указанные ID типов учётных единиц, которые попадают в отчёт
triggerFields	Таблица П.3.1. Приложение 3	Триггерные поля учетных единиц, изменения которых создают запись в отчёте. Используется только для отчетов которые готовятся за период (параметр “usePeriod” = true)
fields		Поля, которые входят в журнал. Структура описания полей отчёта зависит от значений в параметре source. И указана в Таблицах П.3.2, П.3.3. Приложения 3.
postprocessing		Постпроцессинг - действия с отчётом после его формирования. Как правило это фильтрация, сортировка и другие действия над данными отчёта.

		Структура и поддерживаемые типы постпроцессинга описаны в Приложении 3, таблица П.3.4.
permissions		Указывается набор прав для отчёта. Поддерживаемый набор и структура их описания указана в таблице 4.2.

Таблица 4.2. Синтаксис прав для отчёта

Параметр	Описание
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации)}.view",	Право для просмотра отчёта
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации)}.generate"	Право на создание отчёта
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации)}..download"	Право на скачивание отчёта
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации)}.remove"	Право на удаление отчёта
"{Ключ для перевода названия права (перевод ключа отображается согласно значениям из файла локализации)}.send"	Право на отправку отчёта

5. ДЕЙСТВИЯ ПРИ АВАРИЙНЫХ СИТУАЦИЯХ

ПО должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями администратора, неверным форматом или недопустимыми значениями входных данных. В указанных случаях администратору выдаются соответствующие аварийные сообщения, после чего ПО возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных. Аварийные ситуации могут возникать как из-за ошибок в программных продуктах, так и из-за неправильной настройки.

Основными признаками аварийной ситуации являются:

1. Отсутствие на экране необходимой страницы.
2. Окна с сообщениями о нештатной ситуации.

При отказе магнитных носителей или обнаружения ошибок в данных администратор ПО должен восстановить файлы и данные, необходимые для корректной работы ПО из последней резервной копии. Если администратор не может устранить ошибки в данных, следует обратиться к разработчику ПО. При этом необходимо указать перечень данных, содержащих ошибки и правильные значения искаженных атрибутов

В случае возникновения других аварийных ситуаций при работе с ПО и невозможности устранить их с помощью средств администрирования, системы управления базой данных, операционной системы следует обратиться к разработчику ПО. При этом необходимо описать признаки аварийной ситуации и действия, которые были выполнены пользователем непосредственно перед возникновением аварийной ситуации. Ниже описаны основные возможные аварийные ситуации и способы их решения.

Аварийная ситуация	Возможные потери информации	Способ ликвидации последствий	Исполнитель
Отключение питания аппаратных средств	Несохраненные пользователем данные	Повторный ввод и сохранение информации	Пользователь
Выход из строя аппаратных средств (за исключением жесткого диска)	Несохраненные пользователем данные	Повторный ввод и сохранение информации	Пользователь
Сбой при передаче данных	Передаваемая информация	Повторная отправка данных на сервер	Пользователь

Аварийная ситуация	Возможные потери информации	Способ ликвидации последствий	Исполнитель
Отсутствие на экране необходимой страницы	Несохраненные пользователем данные	Перезагрузка страницы кнопкой «Обновить» интернет-браузера; возврат на предыдущую страницу и повторный клик по ссылке на необходимую страницу	Пользователь
Окна с сообщениями о нештатной ситуации	Несохраненные пользователем данные	Выполнить рекомендации, указанные в сообщении, если таковые имеются. При необходимости обратиться к администратору.	Пользователь

Описание работы с valuePath

ValuePath - вид запроса используемый для доступа к полям в пределах пределах одного типа учётной единицы или полями, в которых есть ссылки на другие учётные единицы.

Язык запросов построенный по следующей схеме:
 $\$field2.fieldB.fieldC1...$

\$ - указывает, что ограничение будет использовать запрос в виде valuePath.

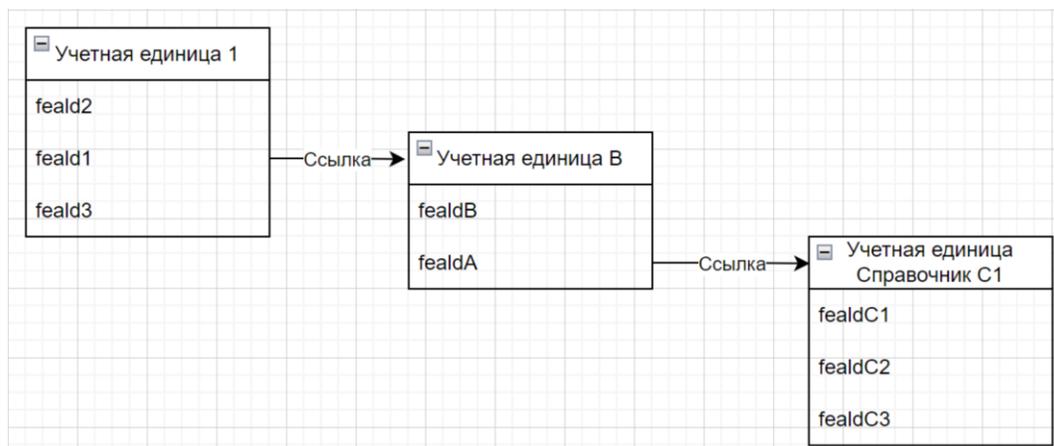
field1 - имя поля в пределах одной учётной единицы (если в поле *field* тип данных является ссылкой на какую либо учётную единицу, то после него ставится точка и указывается необходимое поле *fieldA* в учётной единице, на которую ссылаются).

fieldA - имя поля в учётной единице на которую ссылается *field1*.

fieldC1 - имя поля в учётной единице на который ссылается *fieldA*.

...

Пример: Учетная единица 1 имеет поле *field2*, *feald3* и *field1*, которое ссылается на другую учётную единицу с полями *fealdB*, *fealdA*, в которой *fealdA* ссылается справочник C1. В справочнике C1 есть поле *fieldC1* и *fieldC2*. Необходимо задать ограничение для поля *field1* учётной единицы 1, которое $\geq fieldC1$ из справочника C1, на который ссылается *fealdA*, учётной единицы B.



Результат записи ограничения: **оператор:** “>=” ; **значение:** $\$field1.fieldA.fealdC1$.

Синтаксис поддерживаемых операторов и формул в автозаполняемых полях

Для одного автозаполняемого поля учётной единицы задается одно значение для автозаполнения (выражение). Выражение для автозаполнения может состоять из функций и/или операторов, а также констант.

Аргументами в функциях и для используемых операторов могут быть как константы, так и значения полей, которые указываются в виде `valuePath` (правила указания `valuePath` см. в приложении 1). Значения полей, используемых в выражении, могут быть разных типов данных. Все значения в выражении перед вычислениями приводятся к типу данных первого аргумента.

Например:

- `"abc" + 12 = "abc12"` (строка + число = строка + строка);
- `"1" + "2" = 12` (строка + строка = строка + строка);
- `"" + 4 + 5 = "45"` (пустая строка + число + число = строка + строка + строка);
- `1 + 2 = 3` (число + число = число + число);
- `5 + "6" = 11.` (число + строка = число + число).

При отсутствии значения в одном из полей при расчете выполняется следующая логика:

- строка, текст - `Option.None == ""` (пустая строка);
- целое число и дробное число - `Option.None == 0`;
- дата и время - `Option.None == ошибка`.

После выполнения расчета выражения результат приводится к типу данных поля в котором задано значение для автозаполнения (выражения). Если результат привести невозможно, то система возвращает ошибку проведения расчёта в контрол для автозаполняемого поля и расчётное значение будет отсутствовать.

Автовычисляемое поле, может являться элементом `valuePath` для другого вычисляемого поля. Пример: `fieldTest1 = 5, fieldTest2 = 6, fieldTest3 = 20. fieldTest4 = Sum([$fieldTest1], [$fieldTest2], [$fieldTest3]); fieldTest5= fieldTest4 +29`

- Результат: `fieldTest4 = 31; fieldTest5= 50.`

Перечень поддерживаемых функций и операторов описаны в таблица П1.1 и П1.2.

Таблица П1.1. Операторы

Оператор	Описание	Тип данных результата	Пример
+ (Сложение)	При сложении двух строк происходит континуация (продолжение) строк. Задается в виде: {fild1} + {field2}. Если необходимо указать разделитель между складываемыми полями выражение задается в виде: {fild1} + 'знак разделителя' + {field2}.	Строка	fieldTest1 = Строка1; fieldTest2= Строка2 Пример: [\$fieldTest1] + [\$fieldTest2]. Результат: Строка1Строка2.
	При сложении двух текстовых полей происходит континуация (продолжение). Задается в виде: {text1} + {text2}. Если необходимо указать разделитель между складываемыми полями выражение задается в виде: {text1} + 'знак разделителя' + {text2},	Текст	textTest1 = Текст1; textTest2 = Текст2 Пример: [\$textTest1] + [\$textTest2]. Результат: Текст1Текст2.
	Выполняется сложение целых чисел.	Целое число	fieldTest1 = 4; fieldTest2= 5 Пример: [\$fieldTest1] + [\$fieldTest2]. Результат: 9.
	Выполняется сложение дробных чисел.	Дробное число	fieldTest1 = 4,2; fieldTest2= 5,3 Пример: [\$fieldTest1] + [\$fieldTest2]. Результат: 9,5.
- (Разность)	Выполняется нахождение разности целых чисел.	Целое число	fieldTest1 = 6; fieldTest2= 2 Пример: [\$fieldTest1] - [\$fieldTest2]. Результат: 4.
	Выполняется нахождение разности дробных чисел.	Дробное число	fieldTest1 = 7,4; fieldTest2= 1,3 Пример: [\$fieldTest1] - [\$fieldTest2]. Результат: 6,1.

* (Умножение)	Выполняется умножение целых чисел.	Целое число	fieldTest1 = 6; fieldTest2= 2 Пример: [fieldTest1] * [fieldTest2]. Результат: 12.
	Выполняется умножение дробных чисел.	Дробное число	fieldTest1 = 7,4; fieldTest2= 1,3 Пример: [fieldTest1] * [fieldTest2]. Результат: 9,62.
/ (Деление)	Выполняется деление целых чисел.	Целое число	fieldTest1 = 6; fieldTest2= 2 Пример: [fieldTest1] * [fieldTest2]. Результат: 12.
	Выполняется умножение дробных чисел.	Дробное число	fieldTest1 = 7,4; fieldTest2= 1,3 Пример: [fieldTest1]/ [fieldTest2]. Результат: 9,62.

Таблица П1.2. Функции

Функция	Описание	Поддерживаемые типы данных	Пример
Concat	Функция приводит все аргументы к типу данных "Строка" и выполняет конкатенацию	<ul style="list-style-type: none"> • Строка; • Текст; • Целое число; • Дробное число; • Дата и время; • Дата. 	fieldTest1 = поле с типом данных "Целое число" содержащее значение: 100. fieldTest2 = поле с типом данных "Дробное число" содержащее значение: 4,5. fieldTest3 = поле с типом данных "Строка" содержащее значение: "Мир". fieldTest4 = поле с типом данных "Текст" содержащее значение: "Привет!" Пример: Concat([fieldTest4],[fieldTest3],[fieldTest1],[fieldTest2])

			<p>Результат: Привет!Мир1004,5.</p> <ul style="list-style-type: none"> • Таблица; • Ссылка; • Обратная ссылка. 	<p>fieldTest1 Таблица, которая состоит из 3 записей и полей.</p> <table border="1"> <thead> <tr> <th>element</th> <th>mass</th> </tr> </thead> <tbody> <tr> <td>Уран</td> <td>10.5</td> </tr> <tr> <td>Торий</td> <td>12</td> </tr> <tr> <td>Цезий</td> <td>42</td> </tr> </tbody> </table> <p>fieldTest2 ссылка, на 2 УЕ у которых поле name содержит значения “Привет“ и “Мир“. fieldTest3 обратная ссылка, которая содержит 2 УЕ у которых поле name содержит значения “Ура” и “Круто”.</p> <p>Пример: Concat(selectValues('\$fieldTest1.element'), selectValues('\$fieldTest2.name'), selectValues('\$fieldTest3.name'))</p> <p>Результат: УранТорийЦезийПриветМирУраКруто.</p>	element	mass	Уран	10.5	Торий	12	Цезий	42
element	mass											
Уран	10.5											
Торий	12											
Цезий	42											
Count (Количество элементов)	<p>Нахождение количества элементов в заданном массиве</p> <p>!!!Важно при указании пути не останавливаться на ссылочном поле, а указать какое-нибудь вещественное поле (name, mass, ...).</p>	Таблица	<p>radionuclides таблица, которая состоит их 3 записей.</p> <table border="1"> <thead> <tr> <th>elemet</th> <th>mass</th> </tr> </thead> <tbody> <tr> <td>Уран</td> <td>20</td> </tr> <tr> <td>Торий</td> <td>10</td> </tr> <tr> <td>Цезий</td> <td>45</td> </tr> </tbody> </table> <p>Пример: Count('\$radionuclides.element')</p> <p>Результат: 3.</p>	elemet	mass	Уран	20	Торий	10	Цезий	45	
		elemet	mass									
		Уран	20									
Торий	10											
Цезий	45											
<ul style="list-style-type: none"> • Ссылка • Обратная ссылка 	<p>fieldTest1 поле с обратной ссылкой (или с ссылкой) которая состоит их 5 УЕ.</p> <p>Пример: Count('\$fieldTest1.name')</p> <p>Результат: 5.</p>											
Множественные значения для:	<p>fieldTest1 - строка с 5 множественными значениями.</p>											

		<ul style="list-style-type: none"> • Строка; • Целое число; • Дробное число; • Дата и время; • Дата. 	<p>Пример: Count('\$fieldTest1')</p> <p>Результат: 5.</p>
<p>Pow(x, y)</p> <p>(Возведение в степень)</p>	<p>Выполняется возведение в степень целых чисел.</p>	<p>Целое число</p>	<p>fieldTest1 = 6;</p> <p>Пример: pow([\$fieldTest1],2).</p> <p>Результат: 36.</p>
<p>Где x - это число, а y - степень</p>	<p>Выполняется возведение в степень дробных чисел.</p>	<p>Дробное число</p>	<p>fieldTest1 = 1,4;</p> <p>Пример: pow([\$fieldTest1],0,25.).</p> <p>Результат: 1,0877573059372771726111171278296.</p>
<p>selectValues('\$valuePathToSelect')</p>	<p>Функция для того чтобы вернуть значения типа массив и используется как параметр в других функциях:</p> <ul style="list-style-type: none"> • concat; • sum; • count; • max; • min; • average. <p>Аргумент valuePathToSelect - valuePath поля, где хранится структура\массив ссылок\обратных ссылок.</p>	<ul style="list-style-type: none"> • Таблица • Ссылка • Обратная ссылка 	<p>См. примеры, где используется эта функция как аргумент других функций, перечисленных в описании.</p>
<p>sum</p>		<ul style="list-style-type: none"> • Таблица 	<p>radionuclides = Таблица, которая состоит из 3 записей и полей:</p>

(Сумма)	<p>Нахождение суммы значений в заданном массиве.</p> <p>В функцию может быть передано несколько аргументов, каждый из которых может быть массивом или единственным значением.</p>	<ul style="list-style-type: none"> • Ссылка • Обратная ссылка 	<table border="1"> <thead> <tr> <th>element</th> <th>mass</th> </tr> </thead> <tbody> <tr> <td>Уран</td> <td>10.5</td> </tr> <tr> <td>Торий</td> <td>12</td> </tr> <tr> <td>Цезий</td> <td>42</td> </tr> </tbody> </table> <p>Пример: <code>sum(selectValues('\$radionuclides.mass'))</code></p> <p>Результат: 64,5.</p>	element	mass	Уран	10.5	Торий	12	Цезий	42
	element	mass									
	Уран	10.5									
	Торий	12									
Цезий	42										
	<p>Множественные значения для:</p> <ul style="list-style-type: none"> • Строка с числовым значением; • Целое число; • Дробное число. 	<p>fieldTest1 = поле с типом данных “Целое число” содержащее множественное значение в котором 3 значения: 100; 150; 200. fieldTest2 = поле с типом данных “Дробное число” содержащее множественное значение в котором 3 значения: 4,5; 5,12; 54,43; fieldTest3 = поле с типом данных “Строка” содержащее множественное значение в котором 3 значения: “2”; “5,37”; “165,28”.</p> <p>Пример: <code>Sum([\$fieldTest1], [\$fieldTest2], [\$fieldTest3])</code></p> <p>Результат: 686,7.</p>									
	<p>Единственные значения для:</p> <ul style="list-style-type: none"> • Строка с числовым значением; • Целое число; • Дробное число. 	<p>fieldTest1 = поле с типом данных “Целое число” содержащее значение: 100. fieldTest2 = поле с типом данных “Дробное число” содержащее значение: 4,5. fieldTest3 = поле с типом данных “Строка” содержащее значение “165,28”.</p> <p>Пример: <code>Sum([\$fieldTest1], [\$fieldTest2], [\$fieldTest3])</code></p> <p>Результат: 269,78.</p>									
	<p>Комбинированный случай содержащий множественные и единственные значения в:</p>	<p>fieldTest1 = поле с типом данных “Целое число” содержащее множественное значение в котором 3 значения: 100; 150; 200. fieldTest2 = поле с типом данных “Дробное число” содержащее значение: 4,5. fieldTest3 = поле с типом данных “Строка” содержащее множественное значение в котором 3</p>									

		<ul style="list-style-type: none"> • Строка с числовым значением; • Таблица; • Целое число; • Дробное число. 	<p>значения: “2”; “5,37”; “165,28”. fieldTest4 Таблица, которая состоит из 3 записей и полей</p> <table border="1"> <thead> <tr> <th>element</th> <th>mass</th> </tr> </thead> <tbody> <tr> <td>Уран</td> <td>10.5</td> </tr> <tr> <td>Торий</td> <td>12</td> </tr> <tr> <td>Цезий</td> <td>42</td> </tr> </tbody> </table> <p>Пример: Sum([fieldTest1], [fieldTest2], [fieldTest3],selectValues('fieldTest4.mass'))</p> <p>Результат: 689,65.</p>	element	mass	Уран	10.5	Торий	12	Цезий	42
element	mass										
Уран	10.5										
Торий	12										
Цезий	42										
Ln (Натуральный логарифм)	Нахождение логарифма по основанию e.	<ul style="list-style-type: none"> • Строка (содержащая числовое значение); • Целое число; • Дробное число. 	<p>fieldTest1 = 2.</p> <p>Пример: Ln([fieldTest1])</p> <p>Результат: 0,69314718055994530941723212145818.</p>								
Exp (Функция экспоненты)	Возведение e в степень.	<ul style="list-style-type: none"> • Строка (содержащая числовое значение); • Целое число; • Дробное число. 	<p>fieldTest1 = 0.5.</p> <p>Пример: Exp([fieldTest1])</p> <p>Результат: 1,6487212707001281468486507878142.</p>								
if({условие}, {значение/условие если истина}, {значение/условие если ложь}) (Функция ЕСЛИ)	Функция ЕСЛИ позволяет выполнять логические сравнения значений и ожидаемых результатов. Она проверяет условие и в зависимости от его истинности возвращает результат. Поэтому у функции ЕСЛИ возможны два результата. Первый результат возвращается в случае, если сравнение	<ul style="list-style-type: none"> • Ссылка • Обратная ссылка • Таблица • Строка; • Текст; • Целое число; • Дробное число; • Дата и время; • Дата. 	<p>fieldTest1 = 30, fieldTest2 = Норма, fieldTest3 = Отклонение, fieldTest4.code = 20.</p> <p>Пример: if ([fieldTest1] > [fieldTest4.code],if ([fieldTest1] > 35, [fieldTest2], [fieldTest3]), [fieldTest3])</p> <p>Результат: “Отклонение”.</p>								

	<p>истинно, второй — если сравнение ложно.</p> <p>При записи условия в функции ЕСЛИ можно использовать операторы =, !=, <, <=, >, >=, or, and.</p> <p>В функции ЕСЛИ поддерживается составление сложных выражений, т.е значение истина/ложь может также задаваться с помощью функции ЕСЛИ.</p> <p>Элементами данной функции, могут являться другие функции.</p>		
<p>addDays (Вычисление даты)</p>	<p>Функция для изменения(корректировки) даты на указанное количество дней. Выражение задается в виде:</p> <ul style="list-style-type: none"> • addDays({константа или valuePath}, {константа или valuePath}) - для добавления дней. • addDays({константа или valuePath}, - {константа или valuePath}) - для 	<ul style="list-style-type: none"> • Дата и время; • Дата. 	<p>fieldTest1 = 01.02.2024 18:30</p> <p>Пример: addDays([\$fieldTest1],15)</p> <p>Результат: 16.02.2024 18:30.</p>

	<p>вычета количества дней.</p> <p>В выражении вычисления даты и времени могут использоваться поля только типом дата и время, целое число, дробное число. При наличии в выражении полей других типов, система расчёт не проводит и возвращает ошибку при проведении такого расчёта.</p> <p>Также в выражениях с вычислением даты и времени могут быть заданы константы с размерностью дни.</p>		
<p>addHours (Вычисление времени)</p>	<p>Функция для изменения(корректировки) времени на указанное количество часов. Выражение задается в виде:</p> <ul style="list-style-type: none"> • addHours({констант а или valuePath}, {константа или valuePath}) - для добавления количества часов. • addHours({констант а или valuePath}, - 	<p>•Дата и время</p>	<p>fieldTest1 = 01.02.2024 18:30</p> <p>Пример: addHours([\$fieldTest1], 4)</p> <p>Результат: 01.02.2024 22:30.</p>

	<p>{константа или valuePath}) - для вычета количества часов.</p> <p>В выражении вычисления даты и времени могут использоваться поля только типом дата и время, целое число, дробное число. При наличии в выражении полей других типов, система расчёт не проводит и возвращает ошибку при проведении такого расчёта.</p> <p>Также в выражениях с вычислением даты и времени могут быть заданы константы с размерностью часы.</p>		
<p>dateFormat</p> <p>(Форматирование даты и времени)</p>	<p>Функция для указания формата даты и времени.</p> <p>Выражение задается в виде:</p> <ul style="list-style-type: none"> • dateFormat({константа или valuePath}, '{формат}') <p>В выражении могут использоваться поля</p>	<ul style="list-style-type: none"> • Дата и время; • Дата. 	<p>fieldTest1 = 01.02.2024 18:30</p> <p>Пример: dateFormat([\$fieldTest1], 'dd/MM hh:mm')</p> <p>Результат: 01/02 18:30. (!!!строка)</p>

	<p>только типом дата и время. При наличии в выражении полей других типов, система расчёт не проводит и возвращает ошибку при проведении такого расчёта.</p> <p>!!!Важно форматированная дата будет в часовом поясе сервера.</p> <p>Можно добавить еще один параметр для указания offset (если нужно)</p>		
@now	<p>Функция, которая автоматически заполняет поле текущей датой и текущим временем при открытии страницы.</p>	Дата и время	<p>В fieldTest1 заполнено автозаполняемое поле функцией @now. Пользователь открывает страницу на редактирование с полем fieldTest1 30.05.2024 15:04.</p> <p>Результат: 30.05.2024 15:04.</p>
@today	<p>Функция, которая автоматически заполняет поле текущей датой и временем 00:00 при открытии страницы.</p> <p>Для поля с типом данных "Дата" устанавливается только значение текущей даты.</p>	Дата и время	<p>В fieldTest1 заполнено автозаполняемое поле функцией @today. Пользователь 30.05.2024 15:04 открывает страницу на редактирование с полем fieldTest1.</p> <p>Результат: 30.05.2024 00:00.</p>
		Дата	<p>В fieldTest1 заполнено автозаполняемое поле функцией @today. Пользователь 30.05.2024 15:04 открывает страницу на редактирование с полем fieldTest1.</p> <p>Результат: 30.05.2024.</p>
first	<p>Функция позволяет вывести первый элемент из массива значений. Выводит всегда один элемент если</p>	<ul style="list-style-type: none"> • Строка; • Текст; • Целое число; • Дробное число; 	<p>fieldTest1 = [1, 2, 3]</p> <p>Пример: first('\$fieldTest1')</p>

	<p>будет найдено несколько полей или у одного поля будет несколько значений (поле является массивом)</p>	<ul style="list-style-type: none"> • Дата и время; • Дата. 	<p>Результат: 1.</p> <p>fieldTest1 = [9, 8, 7]; fieldTest2 = [6, 5, 4]; fieldArray = [fieldTest1, fieldTest2]</p> <p>Пример: first('\$fieldArray')</p> <p>Результат: 9.</p>
contains	<p>Функция позволяет проверить содержит ли массив искомое значение.</p> <p>Целесообразно использовать в условиях.</p> <p>1-ый аргумент - valuepath указывающий на массив значений.</p> <p>2-ой аргумент - строка (должен быть в кавычках), наличие которой необходимо проверить в массиве, указанном в первом аргументе</p>	<ul style="list-style-type: none"> • Строка; • Текст; • Целое число; • Дробное число; • Дата и время; • Дата. 	<p>radiationType = ['γ', 'α', 'β']</p> <p>Пример: if(contains('\$type.radiationType', 'γ'), 1, 2)</p> <p>Результат: 1.</p> <p>Пример: if(contains('\$type.radiationType', 'z'), 1, 2)</p> <p>Результат: 2.</p>
ArrayConatains	<p>Функция позволяет проверить содержит ли массив искомое значение.</p> <p>Целесообразно использовать в условиях.</p>	<ul style="list-style-type: none"> • Строка; • Текст; • Целое число; • Дробное число; 	<p>fieldTest1 = 2.</p> <p>Пример: if(arrayContains('[1, 2, 3]', [\$fieldTest1]), 1, 2)</p> <p>Результат: 1.</p>

	<p>1-ый аргумент - массив значений в формате '[1, 2, 3]'. Набор значений разделенных (запятой с пробелом)</p> <p>2-ой аргумент - параметр с valuePath или другая функция</p>		<p>Пример: <code>if(arrayContains('[1, 2, 3]', getEntityDefinitionId('\$name')), 1, 2)</code></p>								
<p>max</p>	<p>Функция позволяет найти максимальное значение из массива целых и дробных чисел.</p> <p>В функцию может быть передано несколько аргументов, каждый из которых может быть массивом или единственным значением.</p>	<ul style="list-style-type: none"> • Таблица • Ссылка • Обратная ссылка • Множественные и единственные значения для: <ul style="list-style-type: none"> ○ Целое число; ○ Дробное число; ○ Строка (содержащая только числовые значения). 	<p>fieldTest1 = поле с типом данных “Целое число” содержащее множественное значение в котором 3 значения: 100; 150; 200. fieldTest2 = поле с типом данных “Дробное число” содержащее значение: 4,5. fieldTest3 = поле с типом данных “Строка” содержащее множественное значение в котором 3 значения: “2”; “5,37”; “165,28”. fieldTest4 Таблица, которая состоит из 3 записей и полей.</p> <table border="1" data-bbox="1396 824 1722 966" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>element</th> <th>mass</th> </tr> </thead> <tbody> <tr> <td>Уран</td> <td>10.5</td> </tr> <tr> <td>Торий</td> <td>12</td> </tr> <tr> <td>Цезий</td> <td>42</td> </tr> </tbody> </table> <p>fieldTest5 ссылка, на 2 УЕ у которых поле number содержит значения 4,2 и 345. fieldTest6 обратная ссылка, которая содержит 2 УЕ у которых поле number содержит значения 32,87 и 23.</p> <p>Пример: <code>max([\$fieldTest1], [\$fieldTest2], [\$fieldTest3], selectValues('\$fieldTest4.mass'), selectValues('\$fieldTest5.number'), selectValues('\$fieldTest6.number'))</code></p> <p>Результат: 345.</p>	element	mass	Уран	10.5	Торий	12	Цезий	42
element	mass										
Уран	10.5										
Торий	12										
Цезий	42										

min	<p>Функция позволяет найти минимальное значение из массива целых и дробных чисел</p> <p>В функцию может быть передано несколько аргументов, каждый из которых может быть массивом или единственным значением.</p>	<ul style="list-style-type: none"> • Таблица • Ссылка • Обратная ссылка • Множественные и единственные значения для: <ul style="list-style-type: none"> ○ Целое число; ○ Дробное число; ○ Строка (содержащая только числовые значения). 	<p>fieldTest1 = поле с типом данных “Целое число” содержащее множественное значение в котором 3 значения: 100; 150; 200. fieldTest2 = поле с типом данных “Дробное число” содержащее значение: 4,5. fieldTest3 = поле с типом данных “Строка” содержащее множественное значение в котором 3 значения: “2”; “5,37”; “165,28”. fieldTest4 Таблица, которая состоит из 3 записей и полей.</p> <table border="1" data-bbox="1396 451 1722 592"> <thead> <tr> <th>element</th> <th>mass</th> </tr> </thead> <tbody> <tr> <td>Уран</td> <td>10.5</td> </tr> <tr> <td>Торий</td> <td>12</td> </tr> <tr> <td>Цезий</td> <td>42</td> </tr> </tbody> </table> <p>fieldTest5 ссылка, на 2 УЕ у которых поле number содержит значения 4,2 и 345. fieldTest6 обратная ссылка, которая содержит 2 УЕ у которых поле number содержит значения 32,87 и 23.</p> <p>Пример: min([\$fieldTest1], [\$fieldTest2], [\$fieldTest3], selectValues('\$fieldTest4.mass'), selectValues('\$fieldTest5.number'), selectValues('\$fieldTest6.number'))</p> <p>Результат: 2.</p>	element	mass	Уран	10.5	Торий	12	Цезий	42
element	mass										
Уран	10.5										
Торий	12										
Цезий	42										
average	<p>Функция позволяет найти среднее значение из массива целых и дробных чисел</p> <p>В функцию может быть передано несколько аргументов, каждый из которых может быть</p>	<ul style="list-style-type: none"> • Таблица • Ссылка • Обратная ссылка • Множественные и единственные значения для: <ul style="list-style-type: none"> ○ Целое число; ○ Дробное число; 	<p>fieldTest1 = поле с типом данных “Целое число” содержащее множественное значение в котором 3 значения: 100; 150; 200. fieldTest2 = поле с типом данных “Дробное число” содержащее значение: 4,5. fieldTest3 = поле с типом данных “Строка” содержащее множественное значение в котором 3 значения: “2”; “5,37”; “165,28”. fieldTest4 Таблица, которая состоит из 3 записей и полей.</p>								

	<p>массивом или единственным значением.</p>	<p>Строка (содержащая только числовые значения).</p>	<table border="1" data-bbox="1398 196 1724 337"> <thead> <tr> <th>element</th> <th>mass</th> </tr> </thead> <tbody> <tr> <td>Уран</td> <td>10.5</td> </tr> <tr> <td>Торий</td> <td>12</td> </tr> <tr> <td>Цезий</td> <td>42</td> </tr> </tbody> </table> <p>fieldTest5 ссылка, на 2 УЕ у которых поле number содержит значения 4,2 и 345. fieldTest6 обратная ссылка, которая содержит 2 УЕ у которых поле number содержит значения 32,87 и 23.</p> <p>Пример: average([\$fieldTest1], [\$fieldTest2], [\$fieldTest3], selectValues('\$fieldTest4.mass'.), selectValues('\$fieldTest5.number'), selectValues('\$fieldTest6.number'))</p> <p>Результат: 78,337142857142857</p>	element	mass	Уран	10.5	Торий	12	Цезий	42
element	mass										
Уран	10.5										
Торий	12										
Цезий	42										

Поддерживаемый синтаксис для конфигурирования отчетов и журналов

Таблица П.3.1. Список поддерживаемых значений в triggerFields

Параметр	Возможные значения	Описание
path	«valuePath»	Триггерное поле в виде записи valuPath, когда триггер срабатывает на каждое изменение поля в valuPath. Количество полей определяет количество уровней в valuPath. Например: "location.location.location.location"
	“{valuePath}\$»	Триггерное поле в виде записи valuPath, когда необходимый триггер срабатывает только на весь путь valuPath. Например: "radionuclideComposition.activity\$"
onlyChanges	true	Триггер срабатывает только на фактическое изменение поля, А -> В. (А -> А, не триггируются).
	false	Триггер срабатывает на любое изменение поля, А -> В, А -> А.

Таблица П.3.2. Список поддерживаемых значений в source

Возможные значения	Описание
instance	Значение указывает на то, что используется поле из учетной единицы. Структура описания поля при source: "instance" следующая: "source": "instance", "fieldName": "системное название поля УЕ, которое необходимо вывести)", "displayName": "отображаемое имя или ключ для перевода графы, в которой отображаются значения поля", "name": "уникальный name (допускается вводить только буквы латинского алфавита) который идентифицирует поле в журнале"

calculated	<p>Значение указывает на то, что поле рассчитывается по определенной формуле. Структура описания поля при source: "calculated" следующая:</p> <pre>"source": "calculated", "formula": "формула для вычисления поля" "displayName": "отображаемое имя или ключ для перевода графы, в которой отображаются значения поля", "name": "уникальный name (допускается вводить только буквы латинского алфавита) который идентифицирует поле в журнале"</pre> <p>Список поддерживаемых формул и операторов в указан таблице П.3.3.</p>
empty	<p>Значение указывает на то, что поле будет пустым и заполнение его при необходимости можно осуществить в постпроцессинге (например постпроцессинг с типом rowNumber). Структура описания поля при source: "empty" следующая:</p> <pre>"source": "empty", "displayName": "отображаемое имя или ключ для перевода графы, в которой отображаются значения поля", "name": "уникальный name (допускается вводить только буквы латинского алфавита) который идентифицирует поле в журнале"</pre>

Таблица П.3.3. Перечень поддерживаемых аргументов в параметре "formula" для "source": "calculated"

Функция	Поддерживаемые типы данных для аргументов функции	Описание
"[\${поле в ссылаемой УЕ} . {название поля в ссылочной УЕ}]"	<ul style="list-style-type: none"> Ссылка 	Вывод значения поля ссылочной УЕ.
"previousValue('\${название поля ссылки} . {название поля в ссылочной УЕ}')" "previousValue('\${название поля}')"	<ul style="list-style-type: none"> Ссылка Обратная ссылка Поле уч.ед. 	Использование предыдущего значения можно указывать путь \$location.location.mass или поле уч.ед \$mass. Будет использовано значение по выбранному пути, которое было на 1мс назад от даты записи.
"join('\${valuePath}', '{раз делитель}')"	<ul style="list-style-type: none"> Ссылка Обратная ссылка Таблица 	Запись значений полученных по заданному пути, через заданный разделитель.

	<ul style="list-style-type: none"> • Множественные значения для: <ul style="list-style-type: none"> ○ Строка; ○ Целое число; ○ Дробное число; ○ Дата и время; ○ Дата. 	Результатом выполнения функции является строка.
<pre>"recursiveDefinitionFind('{название ссылочного поля по которому идет поиск}', '{условие (= или !=)}', {ID типа учётной единицы}, '{поле найденной уч.ед. при истинности условия}', '{значение по умолчанию}', {глубина})"</pre>	<ul style="list-style-type: none"> • Ссылка • Обратная ссылка 	Аргумент, который осуществляет в поле типа ссылка рекурсивный поиск типа учетной единицы по заданному условию, и выводит результат, если результат истина или ложь.
<pre>"recursiveJoin('{название ссылочного поля}', '{название поля, значение которого нужно взять по ссылке}', '{разделитель}', {глубина})"</pre>	<ul style="list-style-type: none"> • Ссылка • Обратная ссылка 	<p>Объединение в одну строку, через заданный разделитель, значений найденных с помощью рекурсии по ссылкам.</p> <p>Если необходимо указать несколько названий поля, можно перечислить через ' '. Пример: нужно отобразить имя местоположения, при этом имена контейнеров это поле name, а имена ячеек - cellName. recursiveJoin('location', 'name cellName', '; ', '5')</p>
<pre>"previousRecursiveJoin('{название ссылочного поля}', '{название поля, значение которого нужно взять по ссылке}', '{разделитель}', {глубина})"</pre>	<ul style="list-style-type: none"> • Ссылка • Обратная ссылка 	<p>Запись в одну строку, через заданный разделитель, значений найденных с помощью рекурсии по ссылкам на момент времени 1мс назад от даты записи.</p> <p>Если необходимо указать несколько названий поля см. recursiveJoin</p>
<pre>"joinArray('\${valuePath}')" "</pre>	<ul style="list-style-type: none"> • Ссылка • Обратная ссылка • Таблица <p>Множественные значения для:</p>	Массив значений полученных по заданному пути.

	<ul style="list-style-type: none"> • Строка; • Целое число; • Дробное число; • Дата и время; • Дата. 	<p>Результатом выполнения функции является массив строк.</p> <p>Отображается в виде строки через разделитель “;”.</p>
<pre>"getOperationInfo('dateTim e code')"</pre>		<p>Функция для использования информации из операции.</p> <p>Возможные параметры:</p> <ul style="list-style-type: none"> • dateTime - возвращает дату и время когда было произведено данное действие • code - возвращает код операции в которой было произведено данное действие
<pre>"getEntityDefinitionId('{ valuePath}')"</pre>	<p>Функция позволяет получить EntityDefinitionId для искомого valuePath.</p> <p>Возвращаемый тип int.</p> <p>В случае если значение не найдено вернется -1.</p> <p>!!!Важно при указании пути не останавливаться на ссылочном поле, а указать какое-нибудь вещественное поле (name, mass, ...).</p>	<pre>getEntityDefinitionId ('\$name')</pre> <p>Результат: EntityDefinitionId текущего Entity, если у него присутствует поле name.</p> <pre>getEntityDefinitionId ('\$location.name')</pre> <p>Результат: EntityDefinitionId того объекта который находится по ссылке location у текущего Entity, если у того присутствует поле name.</p> <pre>getEntityDefinitionId ('\$nonExistingField')</pre> <p>Результат: -1.</p>
<p>Также поддерживаются функции и операторы указанные в приложении 2.</p>		

Постпроцессинг:

- В постпроцессинге используются только поля журнала, не поля учетной единицы.
- Журнальное поле используется в постпроцессинге по характеристике name.

- В постпроцессинге могут быть использованы любые журнальные поля, независимо от их source.
- Все элементы постпроцессинга выполняются последовательно. В том порядке, в котором они заданы в манифесте. Независимо от типа постпроцессинга.

Таблица П.3.4. Описание синтаксиса постпроцессинга и набор используемых аргументов

Параметр	Возможные значения	Описание
Тип hideIdentical		
type	hideIdentical	hideIdentical - скрывание идентичной информации между необходимыми колонками. Обычно используется для журнала перемещений, когда нужно скрыть начальную или конечную часть идентичной информации (например графы Откуда - Куда)
direction	<ul style="list-style-type: none"> • start • finish 	<ul style="list-style-type: none"> • start - скрыть информацию в начале • finish - скрыть информацию с конца.
fieldNames	массив "name" полей, которые участвуют в постпроцессинге	Поля, с характеристикой "name" для которых необходимо скрыть идентичную информацию (Например from и to)
segmentSeparator	Разделитель	Разделитель через который записаны данные в полях участвующих в постпроцессинге
placeholder	Пользовательское значение	Плейсхолдер, который отображается вместо скрытой идентичной информации
Тип filter		
type	filter	filter - отфильтровывает данные в журнале по необходимому значению.
criteria	{ "name постпроцессинга в котором указано поле для фильтрации" : "пользовательское значение" }	Фильтр который выбирает данные с указанным пользовательским значением в поле, у которого задана характеристика "name" для данного постпроцессинга
	{ "name постпроцессинга в котором указано поле	Фильтр который выбирает данные со значением "Да" в булевом поле, у которого задана характеристика "name" для данного постпроцессинга

	для фильтрации}": "true"} {"name постпроцессинга в котором указано поле для фильтрации}": "false"} {"name постпроцессинга в котором указано поле для фильтрации}": "isNull"} {"name постпроцессинга в котором указано поле для фильтрации}": "isNotNull"}	Фильтр который выбирает данные со значением "Нет" в булевом поле, у которого задана характеристика "name" для данного постпроцессинга Фильтр который выбирает данные с пустыми значениями в поле, у которого задана характеристика "name" для данного постпроцессинга Фильтр который выбирает данные с непустыми значениями в поле, у которого задана характеристика "name" для данного постпроцессинга
Тип sort		
type	sort	сортирует данные в журнале по необходимому полю.
fieldNames	"name": "{name поля для сортировки}", "direction": "asc"	Сортировка значений от меньшего к большему (А-Я) в поле, которое задано в характеристике "name"
	"name": "{name поля для сортировки}", "direction": "desc"	Сортировка значений от большего к меньшему (Я-А) в поле, которое задано в характеристике "name"
Тип join		
type	join	заполнение данных в ячейки "исходных" строк журнала, данными из "конечной" строки.
sourceInstanceId	"name" журнального поля отличное от значение "name" в destinationInstanceIds	Поле журнала для исходной УЕ, по которому происходит связь между исходными и конечной строками журнала. В поле должно храниться единичное значение.
destinationInstanceIds	"name" журнального поля отличное от значение "name" в sourceInstanceId	Поле журнала для конечной УЕ, по которому происходит связь между исходными и конечной строками журнала. В поле может храниться как единичное значение, так и массив.
joiningFields	массив "name" журнальных полей	журнальные поля, значения которых будут скопированы из конечной УЕ в исходные УЕ.

merge	true/false	Флаг, будут ли сжаты ячейки (по вертикали) для исходных УЕ. Мерж выполняется только тогда, когда исходные строки располагаются друг за другом.
hideDestinationLine	true/false	Флаг, будет ли скрыта конечная строка из журнала.
overwrite	true/false	Флаг, будет ли скопировано значение из конечной УЕ в исходную в случае, когда ячейка журнала уже заполнена для исходной УЕ.
Тип rowNumber		
type	rowNumber	устанавливает номер строк в выбранное поле.
fieldName	name поля в которое будет установлен номер строки	поле в которое будет установлен номер строки

Теги ClosedXML

Тег	Описание
AddFormulaTag	<p>Тег для заполнения столбца формулой excel.</p> <p>Параметры</p> <p>DeleteNextRow — логическая переменная.</p> <p>DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,</p> <p>DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: **AddFormulaTag{tagId}End**</p> <p>DeleteRowWithTag — логическая переменная, нужна в случае, когда ClosedXML не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.</p> <p>DeleteRowWithTag=true — строка с тегом будет удалена,</p> <p>DeleteRowWithTag=false — строка с тегом не будет удалена. (По умолчанию DeleteRowWithTag=false).</p> <p>Formula — формула excel, которую необходимо вставить в столбцы (см. пример ниже).</p> <p>HeaderKeyWord — заголовок (или его часть) колонки, которую нужно запомнить. Необходим для определения ячейки, с которой начнёт работать тег. Значение должно быть уникальным для всей колонки excel.</p>

NumberOfRowsToHeader — тег для случая, когда между тегом и заголовком больше 1 строки. По умолчанию =1.

TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.

Пример 1

1. Строка с заголовком колонки: $A_i/C_{i,5}$
2. Строка с будущим значением (пустая):
3. Строка с тегом: <<AddFormula
Formula=["INDIRECT(ADDRESS(ROW(),3))/INDIRECT(ADDRESS(ROW(),4))"] DeleteNextRow=true .
HeaderKeyword=["Ai/Ci,5"]>>
4. Строка с флагом окончания действия тега: **AddFormulaTag1End**

	$A_i/C_{i,5}$
<<AddFormula Formula=["INDIRECT(ADDRESS(ROW(),3))/INDIRECT(ADDRESS(ROW(),4))"] DeleteNextRow=true HeaderKeyword=["Ai/Ci,5"]>> **AddFormulaTag1End**	

Результат

A_i^* , Бк/г	$C_{i,5}^{**}$	$A_i/C_{i,5}$
137	0.5	274
60	0.1	600
90	2	45

илионхлила в отходах Бк/г

Пример 2

```
<<AddFormula Formula=["IF(INDIRECT(ADDRESS(ROW(),4))<1,\"Упаковка PAO относится к ОНАО\", \"Упаковка PAO не относится к ОНАО\")"] DeleteNextRow=true HeaderKeyword=["Заключение"] NumberOfRowsToHeader=5 DeleteRowWithTag=true>>
```

16	№	Радионуклид	Удельная активность (A _i), Бк/г	Σ _i A _i /C _i 4*	Критерий отнесения к ОНАО	Заключение
17		{{string.Join(";", Item.Fields.Where(i =>	{{string.Join(";", Item.Fields.Where(i =>			
18						
19	Deleterator=[";"] NumberOfAdditionalColumns="1">>					
20	RowsToHeader=5	**SplitToRowsTag1End**			Σ _i A _i /C _i 3 < 1	
21	Суммарная активность: {{Item.Fields.Where(i => i.Name == "radionuclideComposition").First().Entities.SelectMany(i => i.ZipAndAddFormula)}					
22				ListSeparator=[";"] ItemSeparator=[";"]		<<AddFormula Formula=["IF(INDIRECT(ADDRESS(ROW(),4))<1,\"Упаковка PAO относится к ОНАО\", \"Упаковка PAO не относится к ОНАО\")"] DeleteNextRow=true HeaderKeyword=["Заключение"] NumberOfRowsToHeader=5 DeleteRowWithTag=true>>
23				ZipAction="ZipAndAddFormulaTag1End**		**AddFormulaTag1End**

ClearCellsContentTag Очищает ячейки колонки.

DistinctTag

Тег, который берёт строку в ячейке над ячейкой с тегом, разбивает эту строку в массив, удаляет из него повторяющиеся значения, снова собирает в строку и пишет в ту же ячейку.

DeleteRowWithTag — логическая переменная, нужна в случае, когда **ClosedXML** не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.

DeleteRowWithTag=true — строка с тегом будет удалена,

DeleteRowWithTag=false — строка с тегом не будет удалена. (По умолчанию **DeleteRowWithTag=false**).

	<p>Separator — разделитель, по которому строка разбивается на массив, и по которому элементы массива опять собираются в строку.</p>
FilterFieldTag	<p>Тег для фильтрации строк, работает по ячейкам колонки. Если содержимое ячейки удовлетворяет указанному условию, строка остаётся, если нет — удаляется.</p> <p>Параметры</p> <p>Contains</p> <p>NotContains</p> <p>Примеры</p> <p><<FilterField Contains=["TRUE"]>> (удаляются строки, у которых значение ячейки столбца не равно true)</p> <p><<FilterField NotContains=[""]>> (удаляются строки с пустыми ячейками в столбце)</p>
NumberRowsTag	<p>Тег для нумерации строк таблицы.</p> <p>Параметры</p> <p>DeleteNextRow — логическая переменная.</p> <p>DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,</p> <p>DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: **NumberRowsTag{tagId}End**</p> <p>DeleteRowWithTag — логическая переменная, нужна в случае, когда ClosedXML не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого</p>

поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.

DeleteRowWithTag=true — строка с тегом будет удалена,

DeleteRowWithTag=false — строка с тегом не будет удалена. (По умолчанию DeleteRowWithTag=false).

TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.

В первую ячейку, которая должна быть пронумерована, нужно написать ****NumberRowsTag{tagId}Start****

Примеры

№ п/п
NumberRowsTag1Start
<<NumberRows>>
NumberRowsTag1End

SetToLowerItemsTag

Тег, который идёт по столбцу сверху вниз и перебирает значения ячеек.

Если значение ячейки не равно значению `"*%^$%(#"`, оно [исходное значение] запоминается, а ячейке присваивается `"*%^$%(#"`.

Если значение ячейки равно `"*%^$%(#"`, то ей присваивается последнее запомненное значение, отличное от `"*%^$%(#"`.

Примеры

Место хранения	Дата проверки	Результат проверки
	<code>{{Convert.ToDateTime(Item.FieldValues.Where(i => i.Definition.Name ==</code>	<code>{{Item.FieldValues.Where(i => i.Definition.Name == "inventoryResult").First().Value}}</code>
EntityValue.Field	<code>*\$%^\$%/#</code>	<code>*\$%^\$%/#</code>
<<ColsFit>>	<<FilterField NotContains="">><<SetToLowerItems>>	<<SetToLowerItems>><<LowFilterField NotContains=""*\$%^\$%/#">>

SplitToRowsTag

Тег, который разбивает строку на несколько по указанному разделителю.

Параметры

DeleteNextRow — логическая переменная.

DeleteNextRow=true — строка с флагом окончания действия тега будет удалена,

DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).

Флаг окончания действия тега: `**SplitToRowsTag{tagId}End**`

NumberOfAdditionalColumns — количество колонок справа, значения которых также должны быть разделены на строки по тому же разделителю. Если ячейки слитые, они не считаются за одну (каждая считается отдельно).

Separator — массив разделителей.

TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.

Примеры

1. Строка с заголовком колонки: Наименование СИ

2. Строка с будущим значением: `{{string.Join(";", Item.Fields.Where(i => i.Name == "measuringInstruments").First().Entities.SelectMany(i => . i.Fields).ToList().Where(i => i.Name == "fullName").Select(i => i.Values).SelectMany(i => i).ToArray())}}`
3. Строка с тегом: `<<SplitToRows Separator="[";"]" NumberOfAdditionalColumns="4" Id="1">>`
4. Строка с флагом окончания действия тега: `**SplitToRowsTag1End**`

	A	B	C	D	E	F	G	H
7	Применяемые средства измерения							
8	Наименование СИ			Заводской номер	Номер свидетельства о поверке		Дата следующей поверки	
9	<code>first().Entities.SelectMany(i => i.Fields).ToList().Where(i => i.Name == "measuringInstruments").First().Entities.SelectMany(i => . i.Fields).ToList().Where(i => i.Name == "fullName").Select(i => i.Values).SelectMany(i => i).ToArray())}}</code>							
10	<code>rows Separator="[";"]" NumberOfAdditionalColumns="4" <<ToDateTime Format="["dd.MM.yyyy</code>							
11	<code>**SplitToRowsTag1End**</code>							
12	Идентифицированные радионуклиды							
13	№	Радионуклид	Измеренная удельная активность, Бк/г		Неопределенность измерения, Бк/г		Удельная активность, Бк/г	
14	<code>RowsTagRadionuclide").Select(uctures.SelectMany(i => i.Fields).ToList().Where(i => i.Name == "numberR";"]" NumberOfAdditionalColumns="5" Id="2">></code>							
15	<code>numberR";"]" NumberOfAdditionalColumns="5" Id="2">></code>							
16	<code>RowsTagSplitToRowsTag2End**</code>							

Результат

Применяемые средства измерения				
Наименование СИ		Заводской номер	Номер свидетельства о поверке	Дата следующей поверки
device1si		device1f	device1-number	5/18/2030 2:37:00 PM
device2si		device2f	device2certificate	5/22/2053 4:53:00 PM
device1si		device1f	device1-number	5/18/2030 2:37:00 PM
device2si		device2f	device2certificate	5/22/2053 4:53:00 PM
Идентифицированные радионуклиды				
№	Радионуклид	Измеренная удельная активность, Бк/г	Неопределенность измерения, Бк/г	Удельная активность, Бк/г
1	Cs-137	5		137
2	Co-60			60
3	Sr-90	1		90
4	Cs-137	111	15	546

ToDateTimeTag	<p>Тег для перевода даты и времени в ячейке из одного формата в другой.</p> <p>Параметры</p> <p>TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1.</p> <p>DeleteNextRow — логическая переменная. DeleteNextRow=true — строка с флагом окончания действия тега будет удалена, DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: **ToDateTimeTag{tagId}End**</p> <p>Format — требуемый итоговый формат даты.</p> <p>Примеры</p>
----------------------	--

	<p><<ToDateTime Format=["dd.MM.yyyy"]">></p>
ZipAndAddFormulaTag	<p>Тег, который разделяет строку, полученную из двух массивов, выделяет эти массивы, комбинирует их в один массив, затем добавляет в ячейку формулу с новым массивом.</p> <p>Параметры</p> <p>DeleteNextRow — логическая переменная. DeleteNextRow=true — строка с флагом окончания действия тега будет удалена, DeleteNextRow=false — строка с флагом окончания действия тега не будет удалена. (По умолчанию DeleteNextRow=false).</p> <p>Флаг окончания действия тега: **ZipAndAddFormulaTag{tagId}End**</p> <p>HeaderKeyWord — заголовок (или его часть) колонки, которую нужно запомнить. Необходим для определения ячейки, с которой начнёт работать тег. Значение должно быть уникальным для всей колонки excel. TagId — Id тега, целое число. Должно быть уникальным среди тегов такого же типа (нельзя 2 тега AddFormulaTag с Id=1, можно тег AddFormulaTag и NumberRowsTag с Id=1). По умолчанию Id=1. ListSeparator — разделитель массивов в исходной строке. ItemSeparator — разделитель элементов массивов в исходной строке. NumberOfRowsToHeader — тег для случая, когда между тегом и заголовком больше 1 строки. По умолчанию =1. ZipAction — элементы нового массива будут строками, полученными по принципу {i-й элемент 1-го массива} + {ZipAction} + {i-й элемент 2-го массива}. ListAction — разделитель, через который будут соединены элементы нового массива.</p> <pre>cell.FormulaA1 = string.Join(listAction, firstList.Zip(secondList, (first, second) => first + zipAction + second));</pre> <p>Пример 1</p> <p>1. Строка с заголовком колонки: Расчетное значение $\sum_i A_i/C_i,5^{***}$</p>

2. Строка с будущим значением: `{{string.Join(";", Item.Fields.Where(i => i.Name == "radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i => i.Name == "specificActivityOfEachRadionuclide").Select(i => i.Values).SelectMany(i => i).ToArray())}}*{{string.Join(";", Item.Fields.Where(i => i.Name == "radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i => i.Name == "radionuclide").Select(i => i.Entities).SelectMany(i => i).ToList().SelectMany(i => i.Fields).ToList().Where(i => i.Name == "exemptionRateTable5").Select(i => i.Values).SelectMany(i => i).ToArray())}}`
3. Строка с тегом: `<<ZipAndAddFormula ListSeparator="["*"]" ItemSeparator="[";"]" ZipAction="["/"]" ListAction="["+"]" HeaderKeyword="["Расчетное"]">>`
4. Строка с флагом окончания действия тега: `**ZipAndAddFormulaTag1End**`

```

        Расчетное значение  $\sum_i A_i/C_{i,5}^{***}$ 
        {{{string.Join(";", Item.Fields.Where(i => i.Name ==
"radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i =>
i.Name == "specificActivityOfEachRadionuclide").Select(i => i.Values).SelectMany(i =>
i).ToArray())}}*{{string.Join(";", Item.Fields.Where(i => i.Name ==
"radionuclideComposition").First().EntityStructures.SelectMany(i => i.Fields).ToList().Where(i =>
i.Name == "radionuclide").Select(i => i.Entities).SelectMany(i => i).ToList().SelectMany(i =>
i.Fields).ToList().Where(i => i.Name == "exemptionRateTable5").Select(i =>
i.Values).SelectMany(i => i).ToArray())}}}
<<ZipAndAddFormula ListSeparator="["*"]" ItemSeparator="[";"]" ZipAction="["/"]"
ListAction="["+"]" HeaderKeyword="["Расчетное"]">>
**ZipAndAddFormulaTag1End**

```

Результат

Расчетное значение $\sum_i A_i/C_{i,5}^{***}$
919

Пример 2

<<ZipAndAddFormula ListSeparator="["*"]" ItemSeparator="[";"]" ZipAction="["/"]" ListAction="["+""]" HeaderKeyword="["Σi"]" NumberOfRowsToHeader=5 >

6	№	Радионуклид	Удельная активность (Ai), Бк/г	$\Sigma_i A_i/C_i$ *	Критерий отнесения к ОНАО	Заключение
7		{{string.Join(";", Item.Fields.Where(i => i.Name == "radionuclideComposition").First().Entities.SelectMany(i => i.Fields).ToList()).Where(i => i.Name == "specificActivityOfEachRadionuclide").Select(i => i.Values).SelectMany(i => i).ToArray())}}	{{string.Join(";", Item.Fields.Where(i => i.Name == "radionuclideComposition").First().Entities.SelectMany(i => i.Fields).ToList()).Where(i => i.Name == "specificActivityOfEachRadionuclide").Select(i => i.Values).SelectMany(i => i).ToArray())}}			
8	Deleterator="[";"]" NumberOfAdditionalColumns="1">>					
9	Rows1 **SplitToRowsTag1End**					
10	Суммарная активность: {{Item.Fields.Where(i => i.Name == "radionuclideComposition").First().Entities.SelectMany(i => i.Fields).ToList()).Where(i => i.Name == "specificActivityOfEachRadionuclide").Select(i => i.Values).SelectMany(i => i).ToArray())}}					
11						
12						
13						
14	Расчетная дата распада до уровня освобождения от контроля ⁵ : {{Item.Fields.Where(i => i.Name == "dateExemptionFromControl").First().Values.Any() ? Convert.ToDateTime(Convert.ToStr					

ZipTag

Тег, который разделяет строку, полученную из двух или более массивов, выделяет эти массивы, комбинирует их в один массив, затем добавляет в ячейку строку, полученную из этого нового массива. Тег действует на 1 ячейку, расположенную над ячейкой с тегом.

ListSeparator — разделитель массивов в исходной строке.

ItemSeparator — разделитель элементов массивов в исходной строке.

ZipAction — элементы нового массива будут строками, полученными по принципу {i-й элемент 1-го массива} + {ZipAction} + {i-й элемент 2-го массива}.

ListAction — разделитель, через который будут соединены элементы нового массива.

DeleteRowWithTag — логическая переменная, нужна в случае, когда ClosedXML не удаляет сам строку с тегами. Так происходит, если тег находится вне области, а иногда и если тег находится внутри области (условия этого поведения пока неясны). Перед применением тега нужно убедиться, что строка с тегом не удаляется сама, иначе будет удалена строка со значением.

DeleteRowWithTag=true — строка с тегом будет удалена,

DeleteRowWithTag=false — строка с тегом не будет удалена. (По умолчанию DeleteRowWithTag=false).

NumberOfLists — количество массивов в строке. По умолчанию =2.

Пример

Источники образования	<<Zip ListSeparator=["***"] ItemSeparator=[";"] ZipAction=["/"] ListAction=["; "] NumberOfLists=3>><<Distinct Separator=["; "] DeleteRowWithTag=true>>
-----------------------	--

Ячейка с будущим значением: `{{string.Join(";", Item.Items.SelectMany(i => i.Fields).Where(i => i.Name == "placeOfFormation").SelectMany(i => i.Entities).ToList().SelectMany(i => i.Fields).Where(i => i.Name == "building").SelectMany(i => i.Entities).ToList().SelectMany(i => i.Fields).Where(i => i.Name == "name").Select(i => i.Values.First()).ToArray())}}{{string.Join(";", Item.Items.SelectMany(i => i.Fields).Where(i => i.Name == "placeOfFormation").SelectMany(i => i.Entities).ToList().SelectMany(i => i.Fields).Where(i => i.Name == "room").SelectMany(i => i.Entities).ToList().SelectMany(i => i.Fields).Where(i => i.Name == "name").Select(i => i.Values.First()).ToArray())}}{{string.Join(";", Item.Items.SelectMany(i => i.Fields).Where(i => i.Name == "deliveryDepartment").SelectMany(i => i.Entities).ToList().SelectMany(i => i.Fields).Where(i => i.Name == "name").Select(i => i.Values.First()).ToArray())}}`

Ячейка с тегом: `<<Zip ListSeparator=["***"] ItemSeparator=[";"] ZipAction=["/"] ListAction=["; "] NumberOfLists=3>><<Distinct Separator=["; "] DeleteRowWithTag=true>>`

Результат

Источники образования	building1kks/room1kks/d1; building2kks/room2.2kks/d1 (здание/пом., организация/подразделение)
-----------------------	---

Sort	Сортирует область по столбцу, для которого он указан. Параметры Desc и Asc (Asc – по умолчанию) указывают порядок сортировки. Возможна одновременная сортировка по нескольким столбцам. Сортировка происходит справа налево, то есть сначала сортируется самый правый столбец, затем следующий слева и т.д.
Asc	То же, что и <code><sort asc></code>
Desc	То же, что и <code><sort desc></code>
Group	Создает промежуточные итоги по столбцам, в которых указаны теги итогов (<code><sum></code> и т.д.), группируя их по столбцу, для которого он указан.

	<p>Предварительно область сортируется по всем столбцам, для которых указаны теги Group, Sort, Desc и Asc. Порядок сортировки для тега <group> указывается дополнительным параметром - Desc или Asc (Asc по умолчанию). В случае, если тег <group> указан для нескольких столбцов, промежуточные итоги группируются по всем этим столбцам. Группировка происходит справа налево, то есть сначала итоги группируются по крайнему справа столбцу, для которого указан тег <group>, затем по столбцу с опцией <group> слева от него и т.д.</p> <p>Тег может использоваться без итоговых функций. В этом случае, данные группируются без промежуточных итогов. Для форматирования строк промежуточных итогов и заголовков групп используется форматирование служебной строки области.</p> <p>Параметр MergeLabels вызывает объединение ячеек группы в группируемом столбце.</p> <p>Параметр PlaceToColumn позволяет указать столбец, в который будет помещен заголовок группы.</p> <p>Параметр DisableSubtotals позволяет отключить создание промежуточных итогов для столбца.</p> <p>Параметр WithHeader позволяет создавать заголовок группы при использовании промежуточных итогов. В случае если обнаружен тег <SummaryAbove>, промежуточные итоги размещает над данными.</p> <p>Параметр TotalLabel служит для задания заголовка для промежуточных итогов, например, “Итого по подразделению”.</p> <p>Параметр GrandLabel служит для задания заголовка для промежуточных итогов, например, "Итого".</p>
SummaryAbove	Вспомогательный тег для тега <group>. <SummaryAbove> используется для того, чтобы разместить итоги по группам над данными.
DisableGrandTotal	Запрещает создание всех общих итогов при использовании группировки области с промежуточными итогами.
SUM AVG AVERAGE COUNT COUNTNUMS MAX MIN PRODUCT	Теги итогов. Указание его для столбца вызывает подведение итога по столбцу. Итог вычисляется соответствующей функцией Excel. Итоги размещаются в служебном столбце области. По столбцу можно получить только один вид итога (например, сумму или только среднее). В случае если указано несколько итоговых опций, вычисляется только последний указанный итог.

STDEV VAR	
ColsFit	Вызывает автоматическое выравнивание ширины столбцов по значению в ячейках столбцов всего отчета (если указан для отчета), листа (если указан для листа), колонке всего листа (если указан в первой строке листа), области (если указан для области), колонке области (если указан в строке опций) либо для одной ячейки. В случае если указан для листа и для области на этом листе, вызывается только для листа. Тот же принцип от большего к меньшему применяется и для остальных случаев.
RowsFit	Вызывает автоматическое выравнивание высоты строк по значению в ячейках строк всего отчета (если указан для отчета), листа (если указан для листа), колонке всего листа (если указан в первой строке листа), области (если указан для области), колонке области (если указан в строке опций) либо для одной ячейки. В случае если указан для листа и для области на этом листе, вызывается только для листа. Тот же принцип от большего к меньшему применяется и для остальных случаев.
Delete	Удаляет столбец.