Arenadata™ Streaming

Версия - v1.02

Руководство пользователя по работе с сервисом Nifi

2018 Arenadata™

Оглавление

1 Обзор **3**

2 Поддерживаемые браузеры **4**

3 Терминология **5**

4 Пользовательский интерфейс NiFi **7**

5 Построение потока данных **10**
5.1 Добавление компонентов на рабочую область . . . 10
5.2 Версии компонентов . . . 18
5.3 Изменение версии компонента . . . 21

6 Настройка Процессора **23**
6.1 Вкладка SETTINGS . . . 23
6.2 Вкладка SCHEDULING . . . 25
6.3 Вкладка PROPERTIES . . . 30
6.4 Вкладка COMMENTS . . . 30

7 Пользовательские свойства с использованием Expression Language **36**
7.1 Окно переменных Variables . . . 37
7.2 Ссылка nifi.properties . . . 47

8 Управление потоком данных **51**
8.1 Запуск компонента . . . 51
8.2 Остановка компонента . . . 51
8.3 Настройка компонента . . . 52
8.4 Работа удаленных групп процессов . . . 52

В руководстве приведены сведения для пользователей системы по работе с платформой ADS в части сервиса Nifi – поддержка браузеров, особенности терминалогии, описание пользовательского интерфейса, создание потоков данных, настройка процессора, пользовательские свойства, а так же описаны команды и принцип управления потоками данных.

Руководство может быть полезно администраторам, программистам, разработчикам и сотрудникам подразделений информационных технологий, осуществляющих сопровождение платформы.

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io

Глава 1

Обзор

Сервис **NiFi** – это система потоков данных, основанная на концепциях потокового программирования, которая поддерживает мощные и масштабируемые направленные графы маршрутизации данных, их преобразование и логичную системную медиацию. **NiFi** имеет веб-интерфейс пользователя для работы с потоками данных по их проектированию, управлению, обратной связи и мониторингу.

Сервис может быть сконфигурирован по нескольким параметрам качества обслуживания, таким как устойчивость к потерям и гарантированная доставка, низкая латентность и высокая пропускная способность, а также очередность на основе приоритетов. **NiFi** обеспечивает точное происхождение всех данных: полученных, forked-данных, клонированных при соединении, измененных, отправленных и, наконец, отброшенных при достижении своего настроенного конечного состояния.

Информация о системных требованиях, установке и настройке приведена в документе Руководство администратора по работе с сервисом Nifi.

Important: После установки сервиса NiFi необходимо использовать поддерживаемый веб-браузер для просмотра пользовательского интерфейса

Глава 2

Поддерживаемые браузеры

Сервис **Nifi** поддерживает работу с браузерами, представленными в таблице.

Таблица2.1.: Поддерживаемые браузеры

Браузер	Версия
Chrome	Актуальная и актуальная-1
FireFox	Актуальная и актуальная-1
Edge	Актуальная и актуальная-1
Safari	Актуальная и актуальная-1

Актуальная версия и актуальная-1 означает, что пользовательский интерфейс поддерживается в текущей стабильной версии данного браузера и в его предыдущей версии. Например, если текущий стабильный релиз браузера – *45.X*, то официально поддерживаемыми версиями являются *45.X* и *44.X*.

Для Safari, который выпускает основные версии гораздо реже других браузеров, актуальная версия и актуальная-1 представляют собой два последних выпуска.

Поддерживаемые версии браузера обусловлены возможностями пользовательского интерфейса и используемыми зависимостями, функции UI тестируются в указанных версиях браузеров. При этом пользовательский интерфейс может успешно работать в неподдерживаемых браузерах, но на них не проводится тестирование, и результат работы в них неизвестен. Кроме того, UI разработан как рабочий стол и в настоящее время не поддерживается в мобильных версиях браузеров.

В большинстве сред весь пользовательский интерфейс отображается в браузере. Тем не менее, UI имеет адаптивный дизайн, который позволяет прокручивать экраны по мере необходимости в браузерах меньшего размера или на планшетах. В средах, где ширина браузера меньше *800* пикселей, а высота меньше *600* пикселей, части пользовательского интерфейса могут быть недоступными.

Глава 3

Терминология

В руководстве используется характерная терминология, описание основных понятий приведено в далее.

DataFlow Manager, **DFM** – пользователь **NiFi**, имеющий разрешения на добавление, удаление и изменение компонентов потока данных **NiFi**.

FlowFile – файл потока – представляет собой единый фрагмент данных в **NiFi**, состоящий из двух компонентов: *FlowFile Attributes* – атрибуты и *FlowFile Content* – содержимое. Содержимое – это данные, которые представлены файлом потока. Атрибуты – это характеристики, которые предоставляют информацию или контекст о данных; они состоят из пар ключ-значение. Все FlowFiles имеют следующие стандартные атрибуты:

- *uuid* – уникальный идентификатор для FlowFile;
- *filename* – читаемое человеком имя файла, которое может использоваться при хранении данных на диске или на внешнем сервисе;
- *path* – иерархически структурированное значение, которое можно использовать при хранении данных на диске или на внешнем сервисе с целью, чтобы данные не хранились в одном каталоге.

Processor – процессор – это компонент **NiFi**, используемый для прослушивания входящих данных, извлечения данных из внешних источников, публикации данных во внешних источниках и для маршрутизации, преобразования и извлечения информации из FlowFiles.

Relationship – связь. Каждый процессор имеет ноль или более определенных для него связей. Связи указывают результат обработки файла потока FlowFile. После того как процессор завершает обработку FlowFile, он направляет его в одну из связей. Тогда DFM подключает каждую из связей к другим компонентам для определения, куда файл потока должен пойти далее после каждого потенциального результа обработки.

Connection – соединение. DFM создает автоматизированный поток данных, перемещая компоненты из панели инструментов “Components” на рабочую область, а затем соединяя компоненты через “Connections”. Каждое соединение состоит из одной или нескольких связей. Для каждого проведенного соединения DFM может определить, какие связи следует использовать, что позволяет маршрутизировать данные различными способами в зависимости от результата обработки. Каждое соединение содержит очередь FlowFile и, когда FlowFile переносится в конкретную связь, он добавляется в принадлежащую соответствующему соединению очередь.

Controller Service – контроллер – это точки расширения, которые после добавления и настройки DFM в пользовательском интерфейсе запускаются при старте **NiFi** и предоставляют другим компонентам информацию для использования (например, процессорам или другими контроллерам). Общим контроллером, используемым несколькими компонентами, является *StandardSSLContextService*. Он предоставляет возможность конфигурировать свойства хранилища ключей и доверительного хранилища один раз, а затем повторно использовать эту конфигурацию во всем приложении. Идея заключается в том, что вместо того, чтобы

настраивать данную информацию на каждом процессоре, который может нуждаться в ней, контроллер предоставляет ее любому процессору для использования по мере необходимости.

Reporting Task – задача отчетности. Задачи отчетности выполняются в фоновом режиме для предоставления статистических отчетов о происходящем в инстансе **NiFi**. DFM добавляет и настраивает Reporting Task в пользовательском интерфейсе по желанию. Общими задачами отчетности являются *ControllerStatusReportingTask*, *MonitorDiskUsage*, *MonitorMemory* и *StandardGangliaReporter*.

Funnel – воронка – это компонент **NiFi**, используемый для объединения данных из нескольких соединений в одно.

Process Group – группа процессов. Когда поток данных становится сложным, часто полезно рассмотреть его на более абстрактном уровне. **NiFi** позволяет группировать несколько компонентов, таких как процессоры, в группу процессов. При этом пользовательский интерфейс упрощает работу DFM для соединения нескольких Process Group в логический поток данных, а также позволяет DFM выводить группу процессов для просмотра и управления компонентами в ней.

Port – порт. Потокам данных, созданным с использованием одной или нескольких групп процессов, требуется способ подключения группы процессов к другим компонентам потока данных, что достигается за счет портов. DFM может добавить любое количество входных и выходных портов в группу процессов и присвоить им соответствующие имена.

Remote Process Group (RPG) – группа удаленных процессов. Так же, как данные перемещаются в группу процессов и из нее, порой необходимо перенести данные из одного инстанса **NiFi** в другой. В то время как **NiFi** предоставляет множество различных механизмов для передачи данных из одной системы в другую, удаленные группы процессов часто являются самым простым способом для достижения этой цели.

Bulletin – бюллетень. Пользовательский интерфейс **NiFi** выдает значительный объем мониторинга и обратной связи о текущем состоянии приложения. В дополнение к статистике и текущему статусу о каждом компоненте, компоненты могут сообщать бюллетени. Всякий раз, когда компонент сообщает бюллетень, на нем отображается соответствующий значок. На системном уровне бюллетени отображаются в строке состояния в верхней части страницы. Наведение указателя мыши на значок дает подсказку, показывающую время и критичность бюллетеня (*Debug*, *Info*, *Warning*, *Error*), а также само сообщение. Кроме того, бюллетени со всех компонентов можно просмотреть и отфильтровать на странице "Board Page" в общем меню.

Template – шаблон. Часто поток данных состоит из множества подпотоков, которые могут быть повторно использованы. **NiFi** позволяет DFM выбирать часть потока данных (или весь поток) и создавать шаблон. Шаблону присваивается имя и его можно переместить в рабочую область так же, как и другие компоненты. В результате несколько компонентов могут быть объединены вместе, чтобы сделать более крупный блок для создания потока данных. Шаблоны также можно экспортировать как XML и импортировать в другой инстанс **NiFi**, обеспечивая совместное использование.

flow.xml.gz – все, что DFM помещает в рабочую область пользовательского интерфейса **NiFi**, записывается в реальном времени в один файл с именем *flow.xml.gz*. По умолчанию файл находится в каталоге *nifi/conf*. Любые изменения, сделанные в рабочей области, автоматически сохраняются без необходимости принудительного сохранения посредством кнопки "Save". Кроме того, **NiFi** автоматически создает резервную копию данного файла в каталоге архива при его обновлении, которую можно использовать для настройки отката потока. Для этого необходимо остановить **NiFi**, заменить *flow.xml.gz* на требуемую резервную копию и перезапустить **NiFi**. В кластерной среде следует остановить весь кластер **NiFi**, заменить *flow.xml.gz* одного из узлов и перезапустить данный узел. Затем удалить *flow.xml.gz* из других узлов. После подтверждения, что узел запускается как кластер с одним узлом, запустить остальные узлы. При этом замененная конфигурация потока синхронизируется по всему кластеру. Имя, расположение *flow.xml.gz* и поведение автоматической архивации настраиваемы.

Глава 4

Пользовательский интерфейс NiFi

Пользовательский интерфейс (UI) **NiFi** предоставляет механизмы для создания автоматизированных потоков данных, а также их визуализации, редактирования, мониторинга и администрирования. Пользовательский интерфейс может быть разбит на несколько сегментов, отвечающих за различные функции приложения. Далее представлены некоторые скриншоты с выделенными сегментами UI приложения с подробным описанием основных моментов.

При запуске приложения пользователь может перейти к пользовательскому интерфейсу, выбрав в веб-браузере адрес по умолчанию *http://<hostname>:8080/nifi*. Так как изначально разрешения не настроены, любой пользователь может просматривать и изменять поток данных. Конфигурация настроек приведена в Руководстве администратора по работе с сервисом Nifi.

При первом переходе DFM к пользовательскому интерфейсу отображается пустая рабочая область (Рис.4.1.).

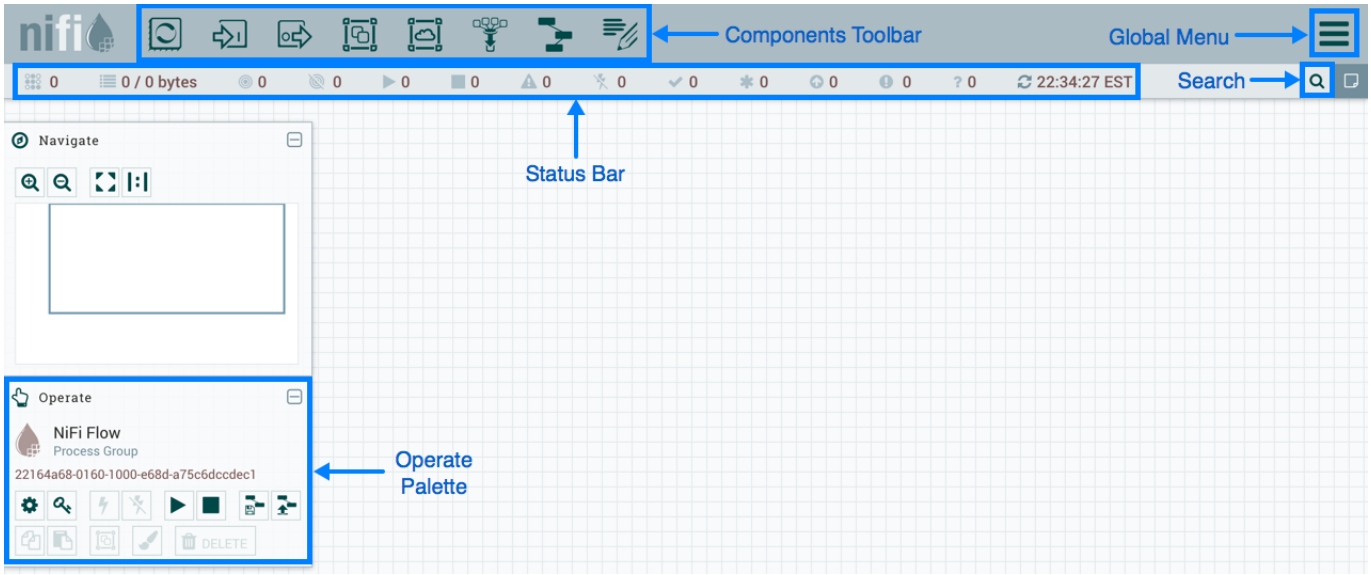


Рис.4.1.: Рабочая область NiFi

Панель инструментов “Components Toolbar” располагается в левой верхней части экрана. Она состоит из компонентов, которые можно переносить в рабочую область для построения потока данных. Каждый компонент панели описан в главе Построение потока данных.

Строка состояния “Status Bar” находится под панелью “Components Toolbar”. Она предоставляет информацию о количестве активных потоков на текущий момент времени, об объеме данных в настоящее время в потоке, количестве групп удаленных процессов в рабочей области в каждом состоянии (*Transmitting*, *Not Transmitting*), сколько процессоров существует в рабочей области в каждом состоянии (*Stopped*, *Running*, *Invalid*, *Disabled*), сколько версий групп процессов существует в рабочей области в каждом состоянии (*Up to date*, *Locally modified*, *Stale*, *Locally modified and stale*, *Sync failure*) и временная метка, в которой вся приведенная информация обновлена в последний раз. Кроме того, если инстанс **NiFi** кластеризован, в строке состояния отображается количество узлов в кластере и сколько из них в настоящее время подключено.

Палитра “Operate” находится в левой части экрана. Она состоит из кнопок управления потоком для DFM, а также для администраторов, управляющих доступом пользователей и настраивающих системные свойства, например, количество предоставленных приложению системных ресурсов.

В правой части рабочей области находится поиск и общее меню. С помощью поиска можно легко находить компоненты в рабочей области по их имени, типу, идентификатору, свойствам конфигурации и их значениям. Общее меню содержит параметры, позволяющие управлять существующими компонентами (Рис.4.2.).

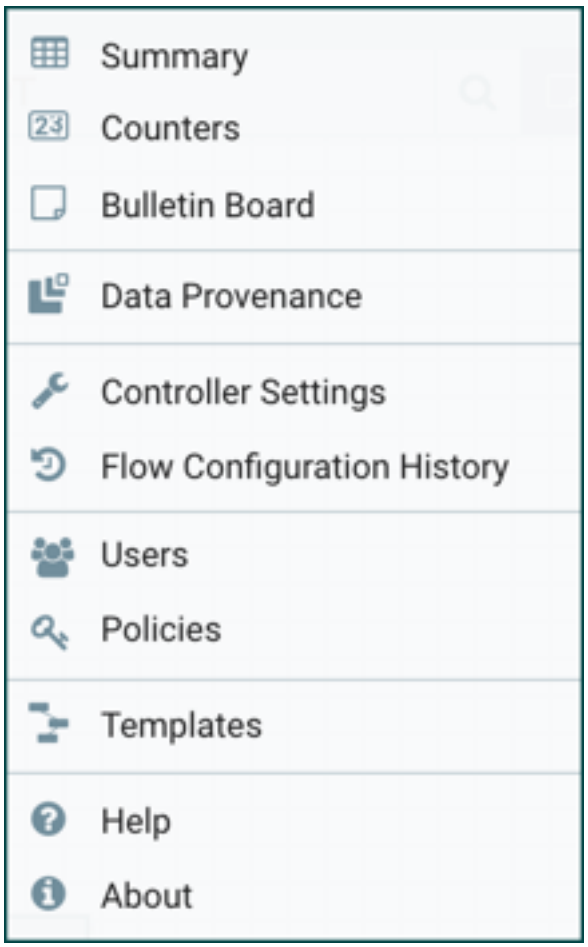


Рис.4.2.: Пармаметры общего меню NiFi

Кроме того, пользовательский интерфейс имеет панель навигации “Navigate”, позволяющую легко перемещаться по рабочей области с возможностью увеличения и уменьшения масштаба. Отдаленный вид потока данных обеспечивает высокоуровневое представление потока и позволяет перемещаться между его крупными частями. Вдоль нижней части экрана располагается навигационная цепочка (“Breadcrumbs”). При переходе в группу процессов цепочка отображают глубину потока и его группы, каждая из которой в свою очередь представляет собой ссылку, по которой можно перейти на тот или иной уровень в потоке (Рис.4.3.).

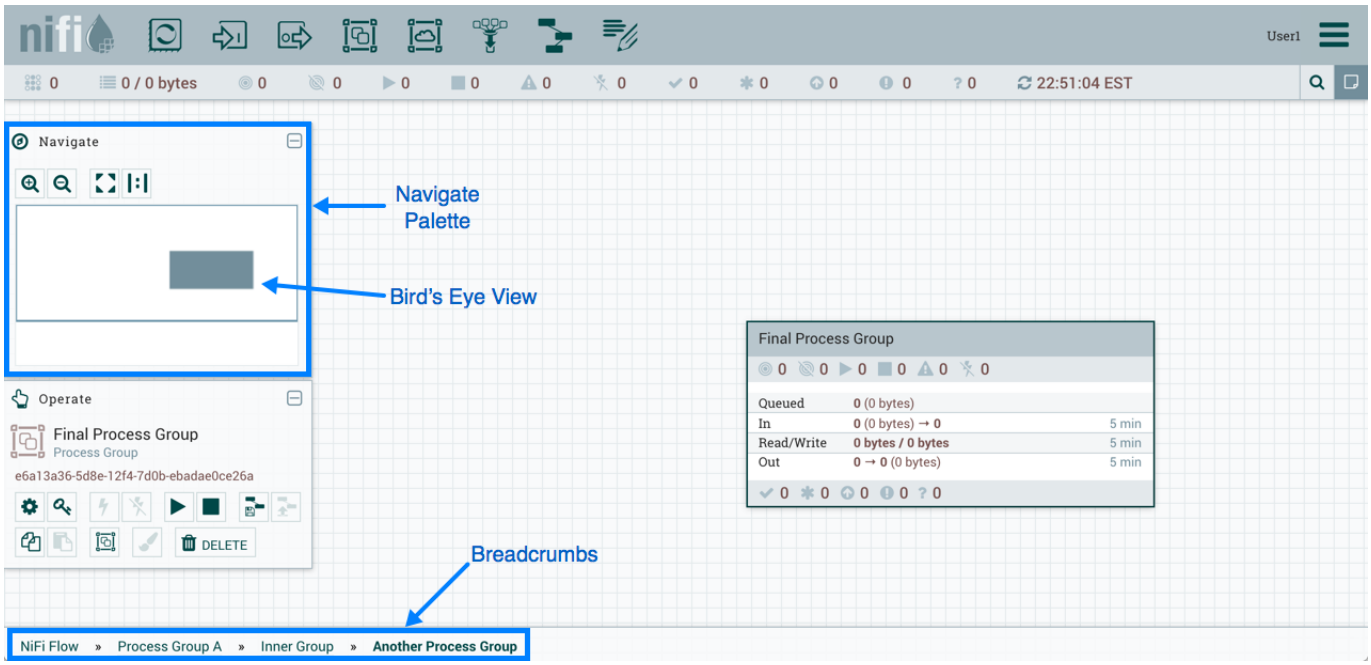


Рис.4.3.: Навигация в NiFi

Глава 5

Построение потока данных

Автоматизированный поток данных создается DFM через пользовательский интерфейс **NiFi**. Для этого необходимо перенести компоненты с панели инструментов на рабочую область, настроить их в соответствии с конкретными потребностями и соединить вместе.

5.1 Добавление компонентов на рабочую область

В разделе Пользовательский интерфейс NiFi описаны различные сегменты UI и упомянута панель инструментов “Components Toolbar”, разбор компонентов которой приведен далее (Рис.5.1.).

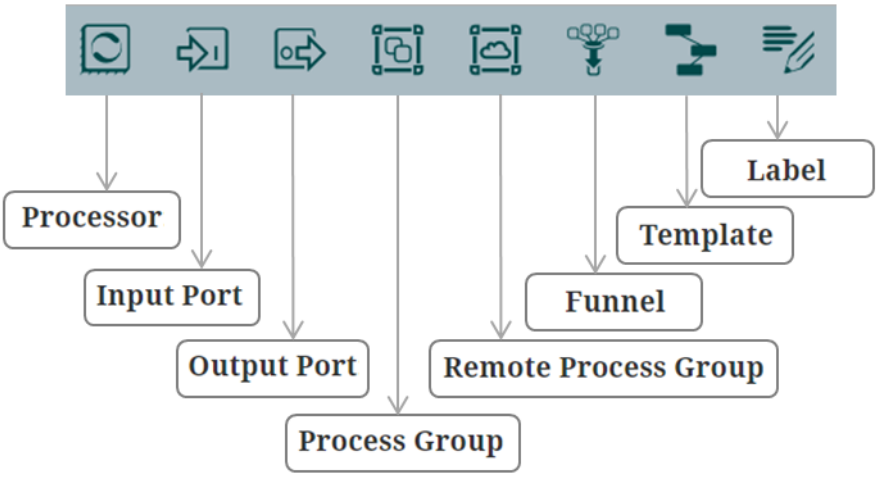
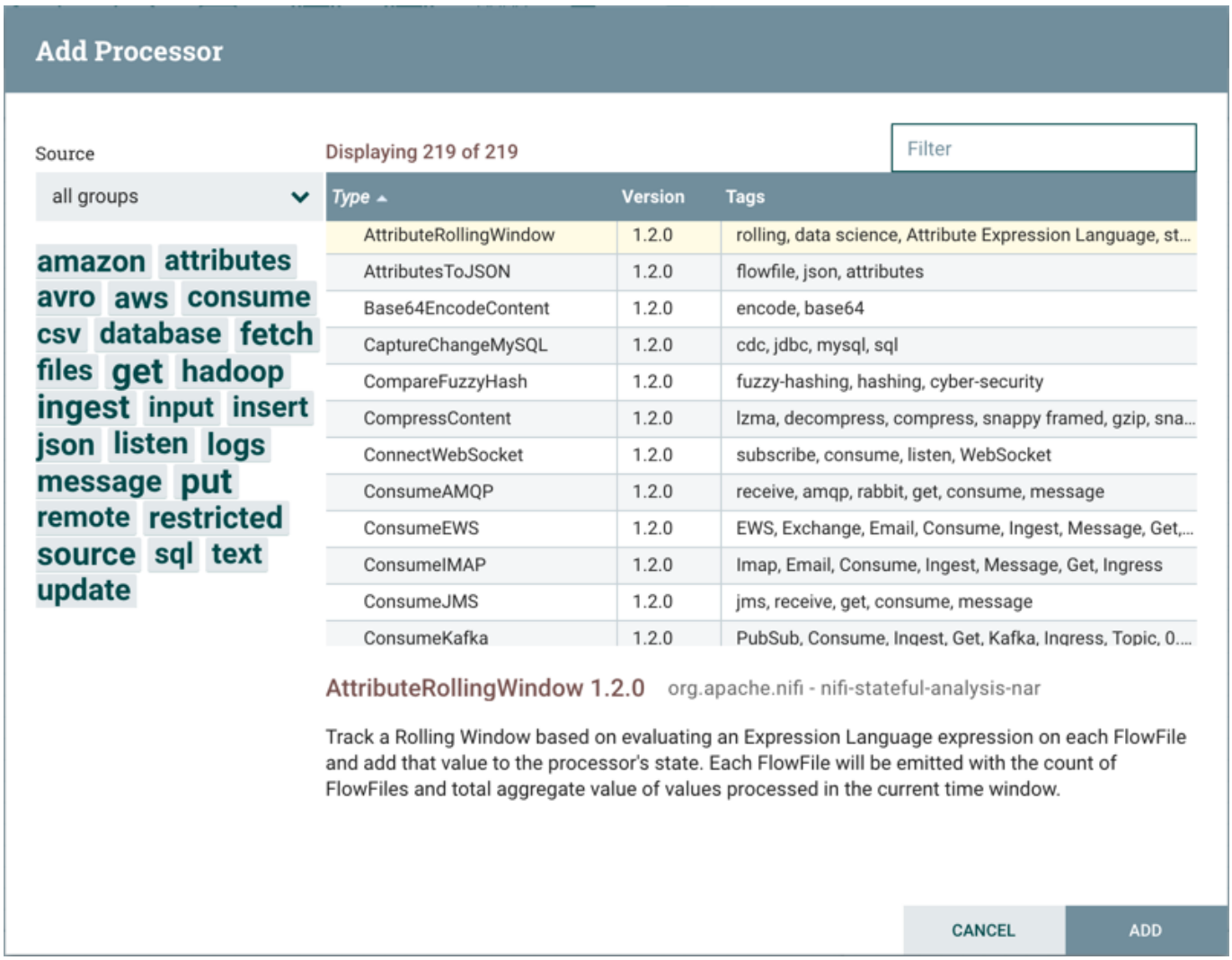


Рис.5.1.: Разбор панели инструментов NiFi

Processor – Процессор является наиболее часто используемым компонентом, поскольку он отвечает за поступление и выход данных, их маршрутизацию и управление. Существуют различные типы Процессоров. Фактически, это очень распространенная точка расширения в **NiFi**, что означает, что поставщики могут реализовать свои собственные Процессоры для выполнения необходимых для их использования функций. При перемещении Процессора на рабочую область **NiFi** открывается диалоговое окно выбора типа (Рис.5.2.).

В правом верхнем углу можно задать фильтр списка на основе типа Процессора или связанных с ним тегов. Разработчики имеют возможность присваивания тегов к своим Процессорам, и в дальнейшем данные



Type	Version	Tags
AttributeRollingWindow	1.2.0	rolling, data science, Attribute Expression Language, st...
AttributesToJSON	1.2.0	flowfile, json, attributes
Base64EncodeContent	1.2.0	encode, base64
CaptureChangeMySQL	1.2.0	cdc, jdbc, mysql, sql
CompareFuzzyHash	1.2.0	fuzzy-hashing, hashing, cyber-security
CompressContent	1.2.0	lzma, decompress, compress, snappy framed, gzip, sna...
ConnectWebSocket	1.2.0	subscribe, consume, listen, WebSocket
ConsumeAMQP	1.2.0	receive, amqp, rabbit, get, consume, message
ConsumeEWS	1.2.0	EWS, Exchange, Email, Consume, Ingest, Message, Get,...
ConsumeIMAP	1.2.0	Imap, Email, Consume, Ingest, Message, Get, Ingress
ConsumeJMS	1.2.0	jms, receive, get, consume, message
ConsumeKafka	1.2.0	PubSub, Consume, Ingest, Get, Kafka, Ingress, Topic, 0....

Рис.5.2.: Выбор типа Процессора

теги отображаются с левой стороны в специальном облаке. Щелчок по тегу в облаке отсортировывает только содержащие данный тег Процессоры. При выборе нескольких тегов отображаются только те Процессоры, которые содержат все указанные теги вкупе. Например, если необходимо показать только те Процессоры, которые позволяют загружать файлы, можно выбрать теги *files* и *ingest* (Рис.5.3.).

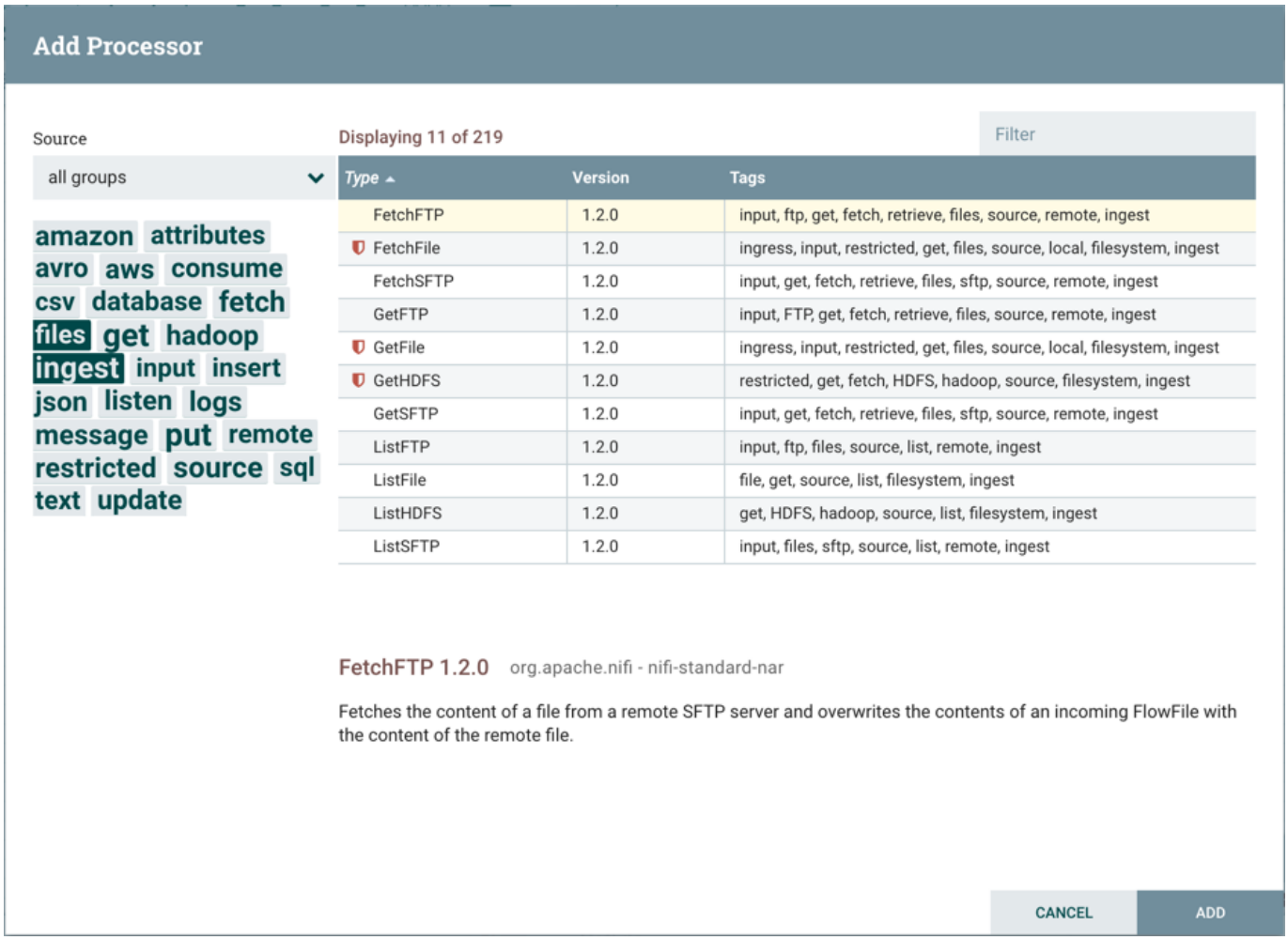


Рис.5.3.: Фильтрация Процессоров по тегам

Ограниченные компоненты отмечаются соответствующим значком слева от названия их типа. Это компоненты, которые могут использоваться для выполнения произвольного необработанного кода, предоставляемого оператором через REST API/UI, или которые могут использоваться для получения или изменения данных в хост-системе **NiFi** с применением учетных данных ОС. Ограниченные компоненты также необходимы авторизованным пользователем **NiFi** с целью выхода за пределы предполагаемого использования приложения, повышения привилегий или предоставления информации о внутренних компонентах процесса **NiFi** или хост-системы.

Все эти возможности следует считать привилегированными, администраторам необходимо знать о них и явно включать для подмножества доверенных пользователей. Но прежде чем пользователь сможет создавать и изменять ограниченные компоненты, ему должен быть предоставлен доступ на определенные разрешения, которые отображаются при наведении курсора на значок “Restricted”. При этом разрешения могут назначаться независимо от ограничений. В таком случае пользователь имеет доступ ко всем ограниченным компонентам. Также пользователям может быть назначен доступ к определенным ограничениям. И в случае если пользователю предоставляется доступ ко всем ограничениям, требуемым компоненту, то он будет иметь доступ к данному компоненту (при условии наличия достаточных разрешений).

Нажатие кнопки “Add” или двойное нажатие на тип Процессора добавляет выбранный Процессор в рабочую область на то место, куда была перемещена пиктограмма с панели инструментов.

Important: Любой компонент в рабочей области можно выбрать с помощью мыши и переместить. Также можно управлять несколькими элементами одновременно – удерживая нажатой клавишу Shift и выбрав каждый компонент

С Процессором в рабочей области можно взаимодействовать, щелкнув правой кнопкой мыши по нему и выбрав параметр из выпадающего контекстного меню. Доступные опции различаются в зависимости от назначенных пользователю привилегий (Рис.5.4.).

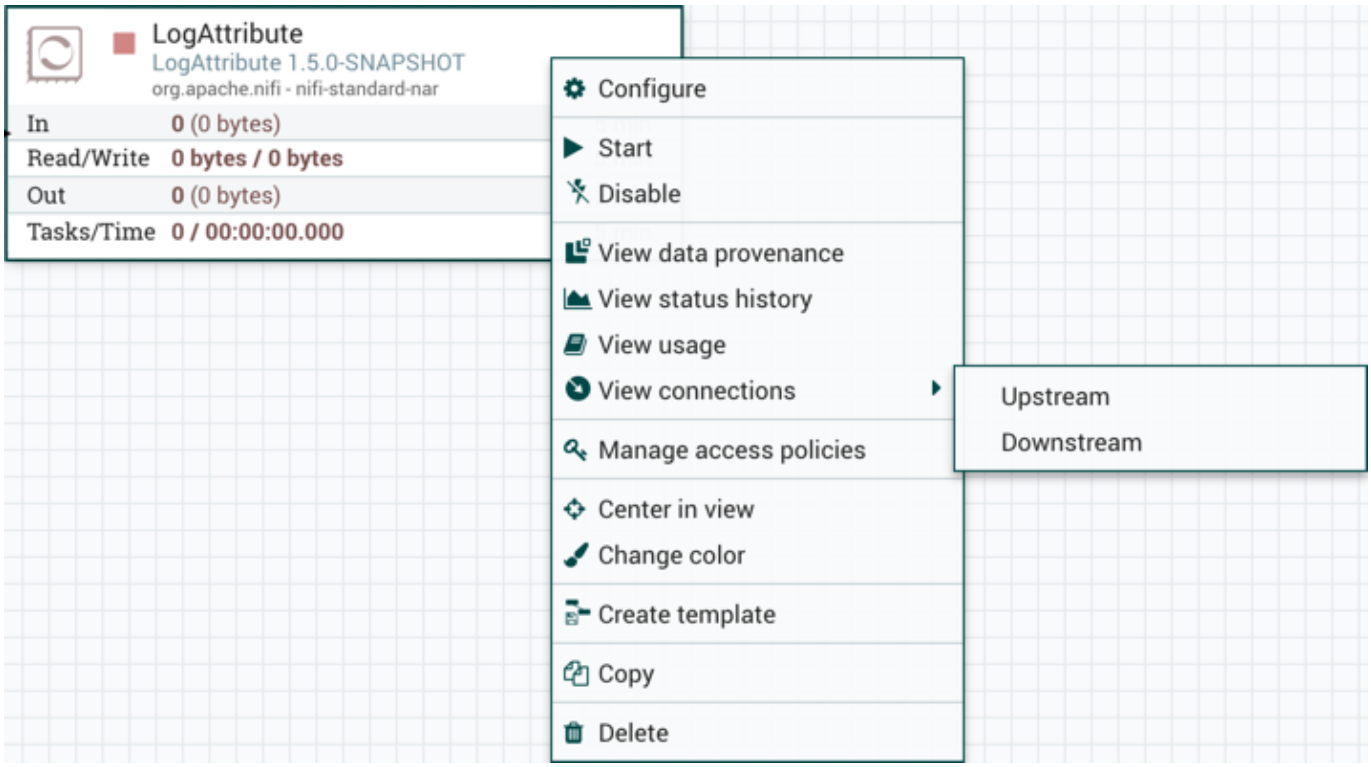


Рис.5.4.: Опции Процессора

Хотя опции контекстного меню различаются, приведенные далее обычно доступны:

- *Configure* – опция позволяет пользователю устанавливать или изменять конфигурацию Процессора (см. Настройка Процессора). Для Процессоров, портов, групп удаленных процессов, соединений и меток диалоговое окно конфигурации можно открыть двойным щелчком по компоненту;
- *Start* или *Stop* – опция позволяет пользователю запускать или останавливать Процессор, в зависимости от его текущего состояния;
- *Enable* или *Disable* – опция позволяет пользователю включить или отключить Процессор, в зависимости от его текущего состояния;
- *View data provenance* – опция отображает таблицу происхождения данных *NiFi Data Prevenance* с информацией о событиях для FlowFiles, проложенных через выбранный Процессор;
- *View status history* – опция открывает графическое представление статистической информации Процессора с течением времени;

- *View usage* – опция позволяет перейти к документации по использованию Процессора;
- *View connections* → *Upstream* – опция позволяет пользователю видеть и переходить на восходящие соединения, входящие в Процессор. Это особенно полезно, когда Процессоры подключаются к другим группам процессов и выходят из них;
- *View connections* → *Downstream* – опция позволяет пользователю видеть и переходить на нисходящие соединения, входящие в Процессор. Это особенно полезно, когда Процессоры подключаются к другим группам процессов и выходят из них;
- *Center in view* – опция центрирует представление рабочей области на данном Процессоре;
- *Change color* – опция позволяет пользователю изменять цвет Процессора, что упрощает визуальный менеджмент больших потоков;
- *Create template* – опция позволяет пользователю создать шаблон из выбранного Процессора;
- *Copy* – опция помещает копию выбранного Процессора в буфер обмена, чтобы можно было его добавить в другое место рабочей области, щелкнув правой кнопкой мыши и выбрав “Paste”. Действия Copy/Paste также могут выполняться с помощью комбинации клавиш “Ctrl-C” (“Command-C”) и “Ctrl-V” (“Command-V”);
- *Delete* – опция позволяет DFM удалять Процессор с рабочей области.

Input Port – Входной порт предоставляет механизм для передачи данных в группу процессов. Когда входной порт перемещается на рабочую область, DFM получает запрос на имя порта. Все порты в группе процессов должны иметь уникальные имена.

Все компоненты существуют в Process Group. Когда пользователь изначально переходит на страницу **NiFi**, он помещается в Root Process Group. Если входной порт перемещается в данную группу процессов, входной порт обеспечивает механизм для приема данных из удаленных инстансов **NiFi** посредством Site-to-Site. В таком случае входной порт может быть настроен для ограничения доступа к соответствующим пользователям при настройке безопасного запуска **NiFi**.

Output Port – Выходной порт предоставляет механизм для передачи данных из группы процессов в места назначения за ее пределами. Когда выходной порт перемещается на рабочую область, DFM получает запрос на имя порта. Все порты в группе процессов должны иметь уникальные имена.

Если выходной порт перемещается в Root Process Group, он обеспечивает механизм отправки данных удаленным инстансам **NiFi** посредством Site-to-Site. В таком случае порт действует как очередь. Поскольку удаленные инстансы **NiFi** извлекают данные из порта, эти данные удаляются из очередей входящих соединений. При настройке безопасного запуска **NiFi** выходной порт можно сконфигурировать для ограничения доступа к соответствующим пользователям.

Process Group – Группы процессов могут использоваться для логического объединения набора компонентов с целью упрощения понимания и управления потоком данных. Когда группа процессов перемещается на рабочую область, у DFM запрашивается имя Process Group, после чего группа процессов вкладывается в родительскую группу. Все Process Group в одной родительской группе должны иметь уникальные имена.

С группой процессов в рабочей области можно взаимодействовать, щелкнув правой кнопкой мыши по ней и выбрав параметр из выпадающего контекстного меню. Доступные опции различаются в зависимости от назначенных пользователю привилегий (Рис.5.5.).

Хотя опции контекстного меню различаются, приведенные далее обычно доступны:

- *Configure* – опция позволяет пользователю устанавливать или изменять конфигурацию группы процессов;
- *Variables* – опция позволяет пользователю создавать или настраивать переменные в пользовательском интерфейсе **NiFi**;

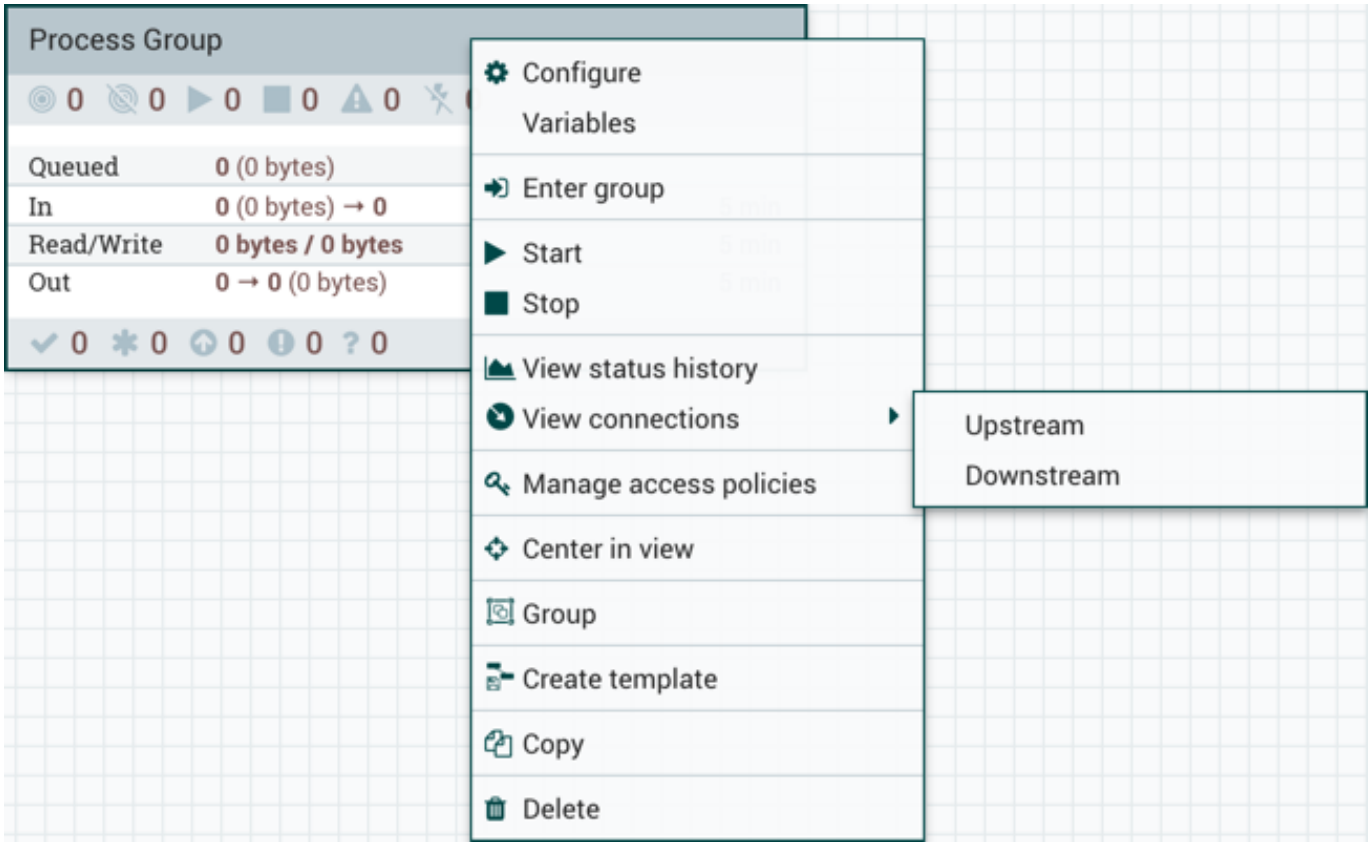


Рис.5.5.: Опции группы процессов

- *Enter group* – опция позволяет пользователю войти в группу процессов. Также можно дважды щелкнуть по группе процессов, чтобы войти в нее;
- *Start* – опция позволяет пользователю запустить группу процессов;
- *Stop* – опция позволяет пользователю остановить группу процессов;
- *View status history* – опция открывает графическое представление статистической информации группы процессов с течением времени;
- *View connections* → *Upstream* – опция позволяет пользователю видеть и переходить на восходящие соединения, входящие в группу процессов;
- *View connections* → *Downstream* – опция позволяет пользователю видеть и переходить на нисходящие соединения, входящие в группу процессов;
- *Center in view* – опция центрирует представление рабочей области на данной группе процессов;
- *Group* – опция позволяет пользователю создать новую группу процессов, содержащую выбранную и любые другие компоненты, указанные на рабочей области;
- *Create template* – опция позволяет пользователю создать шаблон из выбранной группы процессов;
- *Copy* – опция помещает копию выбранной группы процессов в буфер обмена, чтобы можно было ее добавить в другое место рабочей области, щелкнув правой кнопкой мыши и выбрав “Paste”. Действия Copy/Paste также могут выполняться с помощью комбинации клавиш “Ctrl-C” (“Command-C”) и “Ctrl-V” (“Command-V”);
- *Delete* – опция позволяет DFM удалять группу процессов с рабочей области.

Remote Process Group – Группы удаленных процессов отображаются и ведут себя аналогично группам процессов. Только группа удаленных процессов (RPG) ссылается на удаленный инстанс **NiFi**. Когда RPG перемещается на рабочую область, у DFM запрашивается URL-адрес удаленного инстанса. Если удаленный **NiFi** является кластеризованным, URL-адрес, который должен использоваться, – это URL-адрес любого инстанса **NiFi** в этом кластере. Когда данные передаются кластеризованному **NiFi** через RPG, RPG подключается к удаленному инстансу, URL-адрес которого настроен для определения, какие узлы находятся в кластере и насколько занят каждый из них. Эта информация используется для балансировки загрузки данных на каждый узел. Затем удаленные инстансы периодически опрашиваются для определения сведений о узлах, которые удаляются из кластера или добавляются в него, и при этом балансировка загрузки каждого узла перерасчитывается.

С удаленной группой процессов в рабочей области можно взаимодействовать, щелкнув правой кнопкой мыши по ней и выбрав параметр из выпадающего контекстного меню. Доступные опции различаются в зависимости от назначенных пользователю привилегий (Рис.5.6.).

Хотя опции контекстного меню различаются, приведенные далее обычно доступны:

- *Configure* – опция позволяет пользователю устанавливать или изменять конфигурацию группы удаленных процессов;
- *Enable transmission* – опция активирует передачу данных между инстансами **NiFi**;
- *Disable transmission* – опция отключает передачу данных между инстансами **NiFi**;
- *View status history* – опция открывает графическое представление статистической информации группы удаленных процессов с течением времени;
- *View connections* → *Upstream* – опция позволяет пользователю видеть и переходить на восходящие соединения, входящие в группу удаленных процессов;
- *View connections* → *Downstream* – опция позволяет пользователю видеть и переходить на нисходящие соединения, входящие в группу удаленных процессов;
- *Refresh remote* – опция обновляет представление состояния удаленного инстанса **NiFi**;

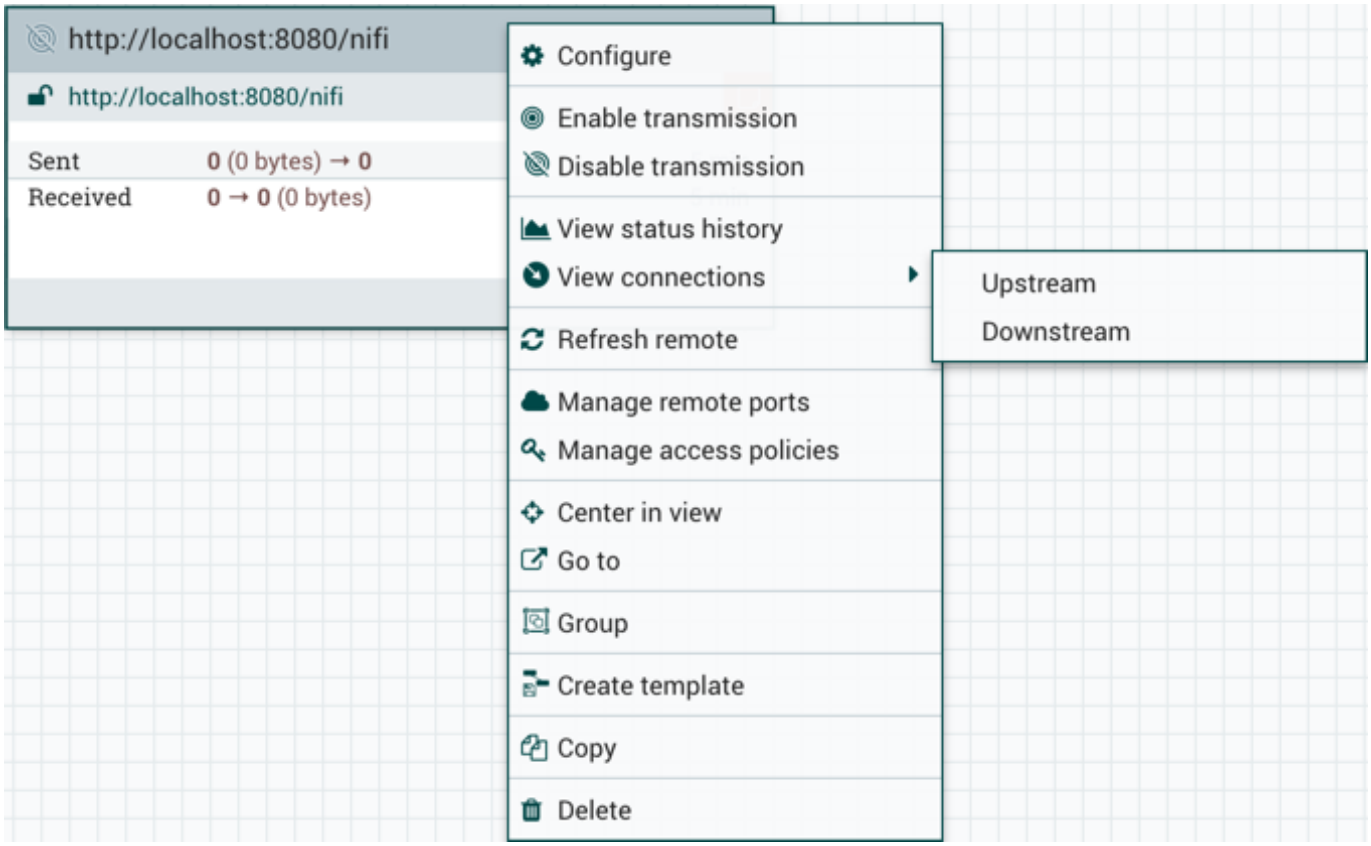


Рис.5.6.: Опции удаленной группы процессов

- *Manage remote ports* – опция позволяет пользователю видеть входные и/или выходные порты, существующие на удаленном инстансе **NiFi**, к которому подключена группа удаленных процессов. При этом если конфигурация Site-to-Site защищена, отображаются только те порты, к которым предоставлен доступ данному пользователю **NiFi**;
- *Center in view* – опция центрирует представление рабочей области на данной группе удаленных процессов;
- *Go to* – опция открывает представление удаленного инстанса **NiFi** на новой вкладке браузера. При этом если конфигурация Site-to-Site защищена, у пользователя должен быть доступ к удаленному инстансу **NiFi** для его просмотра;
- *Group* – опция позволяет пользователю создать группу процессов, содержащую выбранную группу удаленных процессов;
- *Create template* – опция позволяет пользователю создать шаблон из выбранной группы удаленных процессов;
- *Copy* – опция помещает копию выбранной группы удаленных процессов в буфер обмена, чтобы можно было ее добавить в другое место рабочей области, щелкнув правой кнопкой мыши и выбрав “Paste”. Действия Copy/Paste также могут выполняться с помощью комбинации клавиш “Ctrl-C” (“Command-C”) и “Ctrl-V” (“Command-V”);
- *Delete* – опция позволяет DFM удалять группу удаленных процессов с рабочей области.

Funnel – Воронки используются для объединения данных из нескольких Соединений в одно, что имеет два преимущества. Во-первых, при наличии большого количества Соединений с одним и тем же назначением рабочая область может загромождаться занимаемым ими пространством. Путем объединения Соединений в одно, полученное одиночное Соединение затем можно так же нарисовать на рабочей области, охватив такое же пространство. Во-вторых, Соединения могут быть настроены с помощью приоритетов FlowFile. Данные из нескольких Соединений могут быть направлены в одиночное Соединение, обеспечивая возможность приоритизации всех данных, а не определять приоритеты данных по каждому Соединению независимо друг от друга.

Template – Шаблоны могут создаваться DFM из части потока или могут импортироваться из других потоков данных. Они обеспечивают крупные блоки для быстрого создания сложного потока. При перемещении пиктограммы “Template” на рабочую область открывается диалоговое окно для выбора шаблона из списка доступных (Рис.5.7.).

В раскрывающемся списке находятся все доступные шаблоны. Любой шаблон, созданный с описанием, содержит значок вопроса, указывающий на наличие дополнительных сведений, отображающихся при наведении курсора мыши на иконку (Рис.5.8.).

Label – Ярлыки используются для предоставления информативного текста частям потока данных. При перемещении пиктограммы “Label” на рабочую область он создается с заданным по умолчанию размером с возможностью последующего редактирования при помощи маркера в правом нижнем углу. Ярлык не имеет текста при создании. Текст добавляется по щелчку правой кнопкой мыши на ярлыке и выбору параметра *Configure*.

5.2 Версии компонентов

В приложении есть доступ к информации о версии Процессоров, контроллера и задач отчетности. Это особенно полезно при работе в кластерной среде с несколькими инстансами **NiFi**, использующими разные версии компонентов, или при обновлении до более новой версии процессора. Диалоговые окна “Add Processor”, “Add Controller Service” и “Add Reporting Task” содержат столбец с версией компонента, а также имя компонента, организации или группы, создавшей его, и содержащий данный компонент пакет NAR (Рис.5.9.).

Каждый компонент на рабочей области также содержит эту информацию (Рис.5.10.).

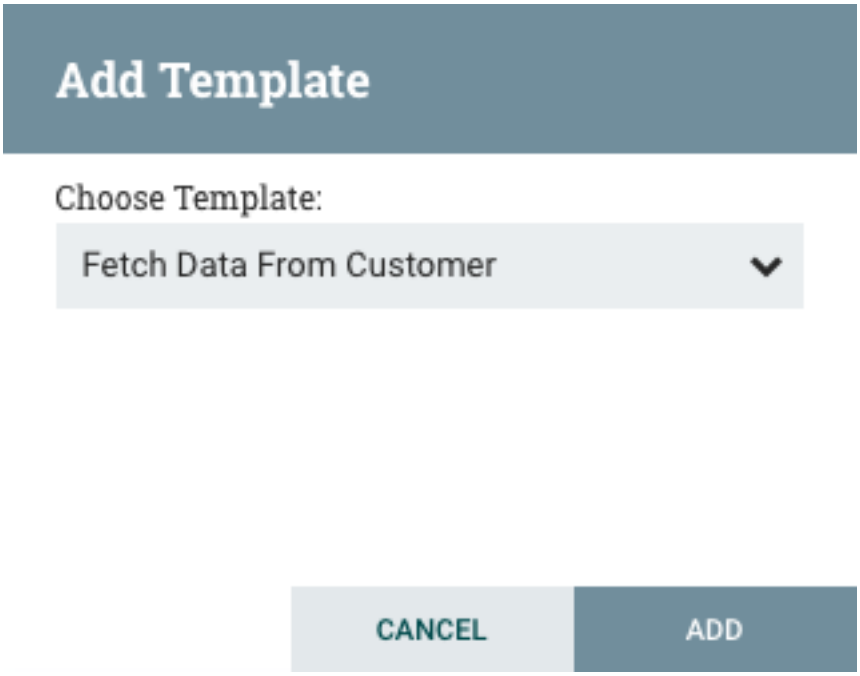


Рис.5.7.: Выбор шаблона

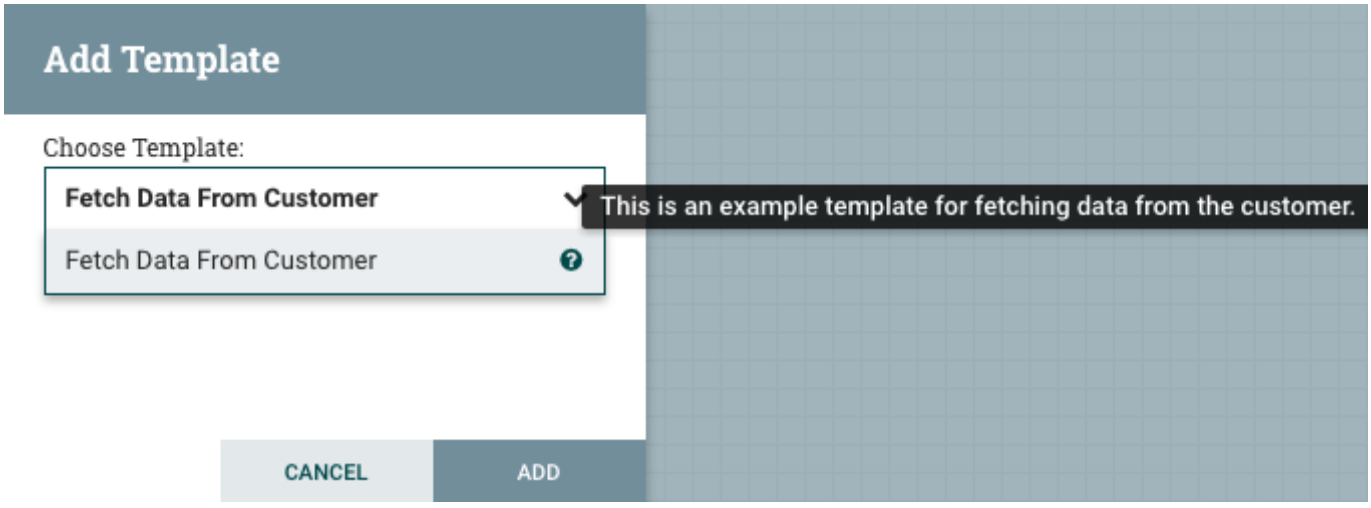


Рис.5.8.: Дополнительные сведения о шаблоне

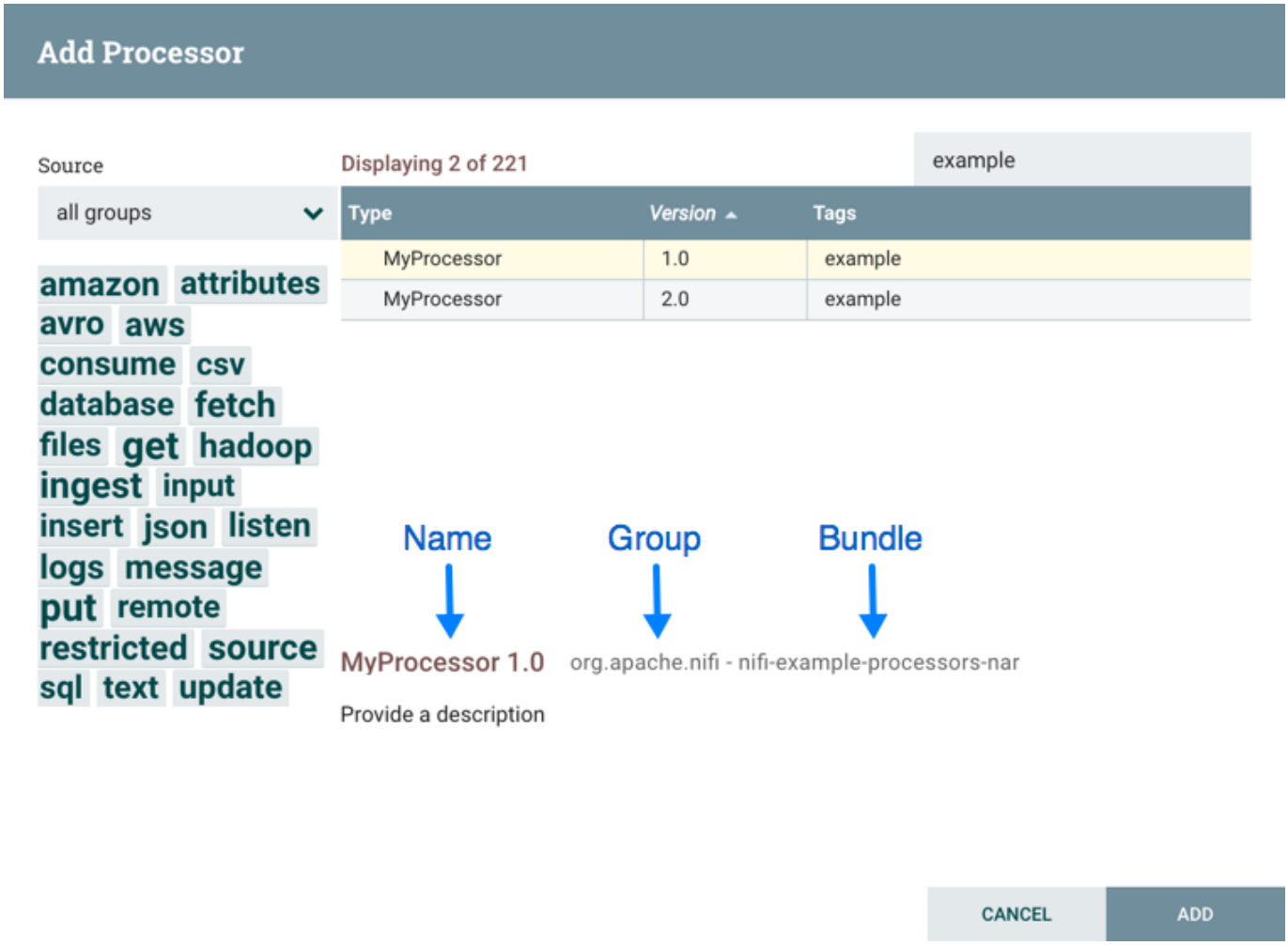


Рис.5.9.: Версии компонентов

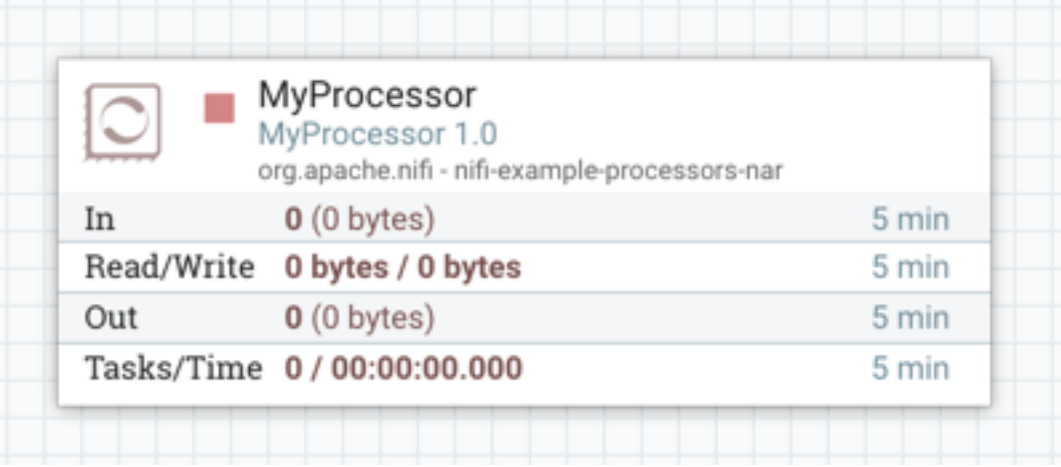


Рис.5.10.: Версия компонента

5.3 Изменение версии компонента

Для изменения версии компонента необходимо выполнить следующие действия:

1. Кликнуть правой кнопкой мыши на компонент в рабочей области для отображения параметров конфигурации.
2. Выбрать “Change version” (Рис.5.11.).

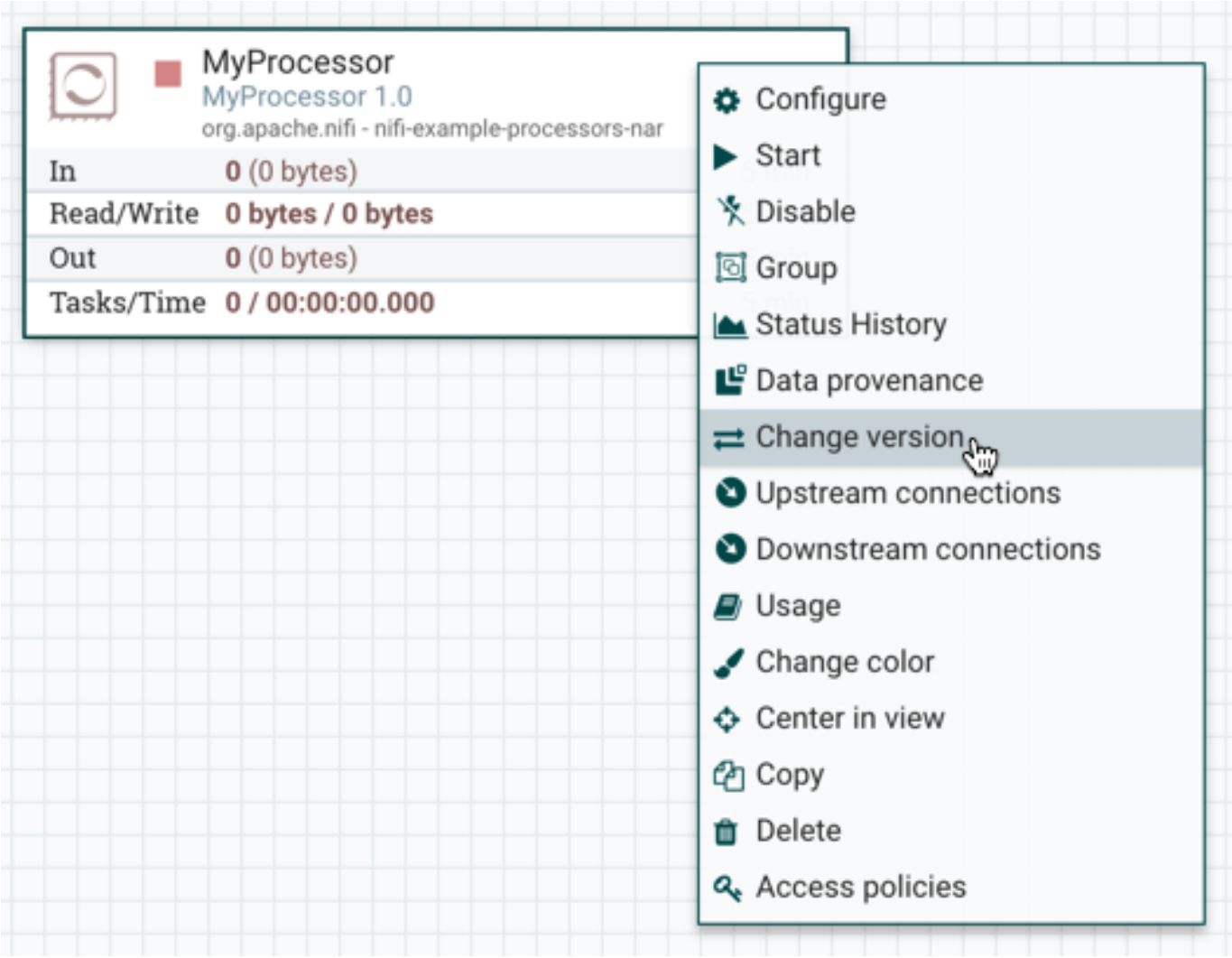


Рис.5.11.: Параметры конфигурации компонента

3. В диалоговом окне “Component Version” выбрать необходимую версию в раскрывающемся меню “Version” (Рис.5.12.).

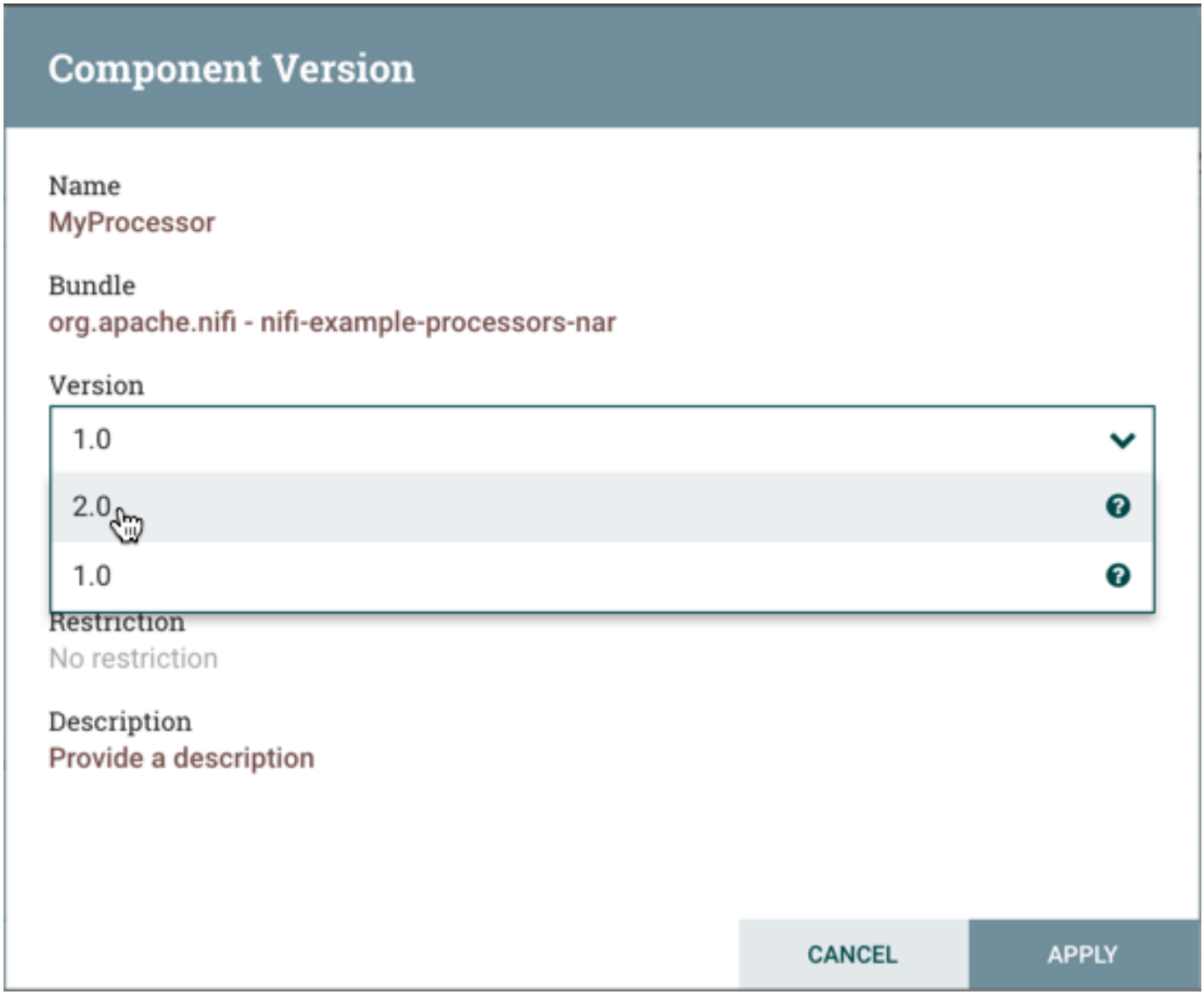


Рис.5.12.: Выбор версии компонента

Глава 6

Настройка Процессора

Окно настройки Процессора открывается по двойному щелчку по Процессору в рабочей области либо при выборе опции “Configure” из контекстного меню Процессора. Диалоговое окно конфигурации имеет четыре вкладки, каждая из которых описана далее. После завершения настройки Процессора для применения изменений следует нажать кнопку “Apply”; для выхода из окна без сохранения – кнопку “Cancel”.

При запущенном Процессоре в его контекстном меню опция “Configure” меняется на “View Configuration”, так как нельзя менять конфигурацию Процессора во время его работы. Для этого необходимо сначала остановить Процессор и дождаться завершения всех его активных задач.

Important: Ввод определенных символов не поддерживается и автоматически отфильтровывается при вводе. Следующие символы и любые непарные суррогатные коды Unicode не сохраняются ни в одной конфигурации:

```
[#x0], [#x1], [#x2], [#x3], [#x4], [#x5], [#x6], [#x7], [#x8], [#xB], [#xC], [#xE], [#xF], [#x10], [#x11], [
↪#x12], [#x13], [#x14], [#x15], [#x16], [#x17], [#x18], [#x19], [#x1A], [#x1B], [#x1C], [#x1D], [#x1E], [
↪#x1F], [#xFFFE], [#xFFFF]
```

6.1 Вкладка SETTINGS

При обращении к диалоговому окну настроек Процессора, оно открывается на первой вкладке “SETTINGS” (Рис.6.1.).

Вкладка содержит несколько элементов конфигурации. В поле “Name” у DFM есть возможность изменить имя Процессора. По умолчанию оно совпадает с типом. Рядом с полем имени располагается флажок “Enabled”, указывающий статус Процессора. При добавлении Процессора на рабочую область он активируется, и если его отключить, дальнейший запуск уже невозможен. Отключенное состояние используется для указания того, что, например, при запуске группы Процессоров данный (отключенный) Процессор следует исключить.

Ниже отображается уникальный идентификатор Процессора в поле “Id”, его тип в поле “Type” и NAR в поле “Bundle”. Данные значения не могут быть изменены.

Далее находятся поля “Penalty Duration” и “Yield Duration”. Во время обычного процесса обработки потока данных (FlowFile) может произойти событие, указывающее, что данные могут быть обработаны не в данный момент, а позднее. Например, если Процессор должен передать данные удаленному сервису, но у сервиса уже есть одноименный файл. В таком случае Процессор может “оштрафовать” FlowFile сроком ожидания, что останавливает обработку данных потока на период времени, заданный в поле “Penalty Duration” (по умолчанию *30 секунд*).

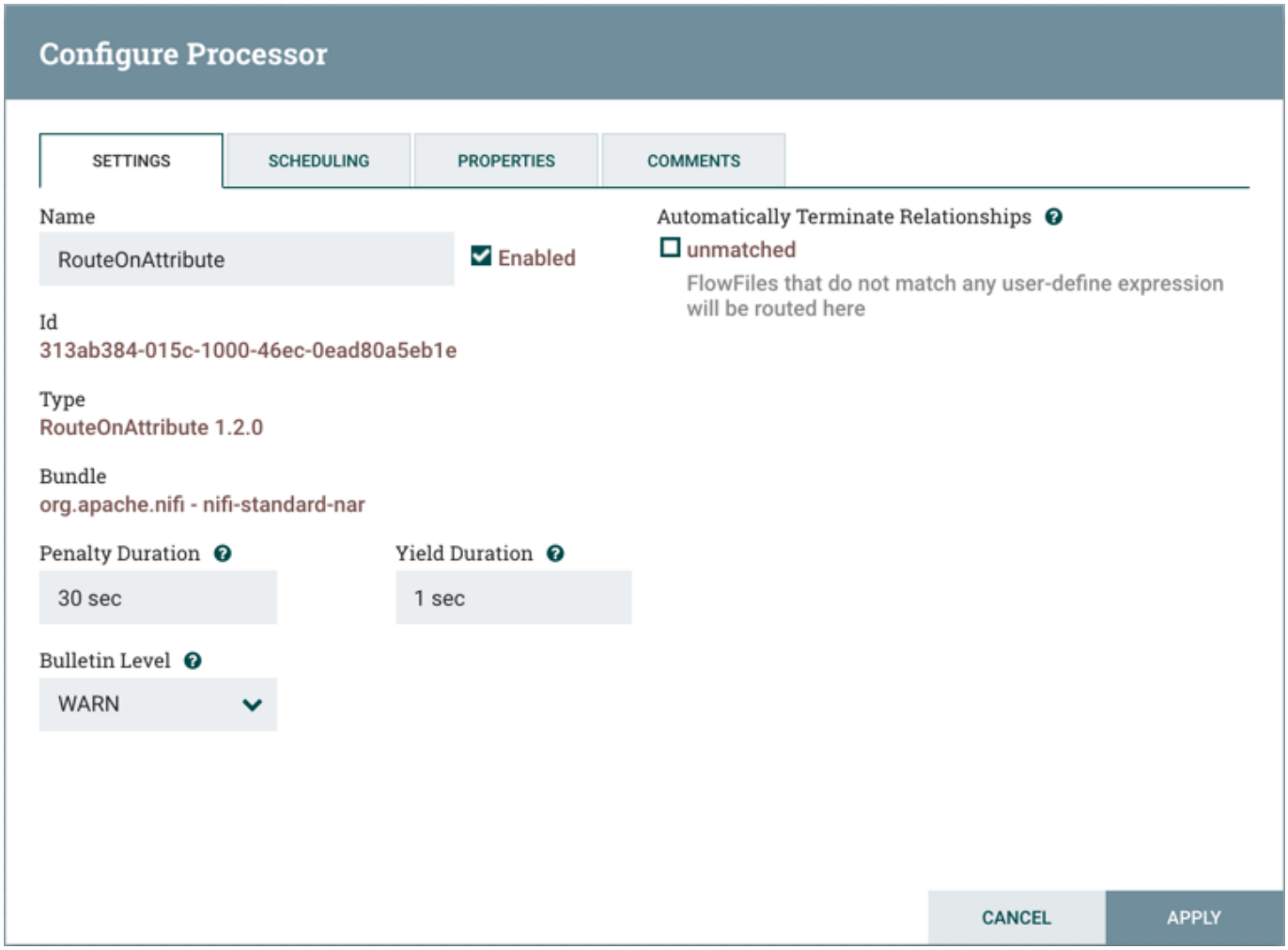


Рис.6.1.: Вкладка SETTINGS

Аналогичным образом Процессор может определить, что существует ситуация, при которой он больше не может выполнять действия независимо от обрабатываемых данных. Например, если Процессор должен отправить данные в удаленный сервис, но сервис не отвечает, Процессор при этом не может добиться какого-либо прогресса. В результате Процессор должен "уступить", а это отодвигает запланированный запуск Процессора на некоторый период времени, который задается в поле "Yield Duration" (значение по умолчанию *1 секунда*).

Последний настраиваемый параметр в левой части вкладки – "Bulletin Level". Всякий раз, когда Процессор записывает данные в свой журнал, он также создает бюллетень. В параметре указывается самый низкий уровень бюллетеня. По умолчанию установлено значение *WARN*, означающее, что в пользовательском интерфейсе отображаются все бюллетени с предупреждениями и ошибками.

В правой части вкладки "SETTINGS" находится "Automatically Terminate Relationships" с перечислением всех определенных Процессором связей и их описанием. Для того чтобы Процессор считался действительным и способным к запуску, каждая определенная им связь должна быть подключена к нисходящему компоненту, в противном случае она автоматически прерывается. При автоматическом прерывании связи любой направляемый в нее FlowFile удаляется из потока, и его обработка считается завершенной. Любая связь, которая уже подключена к нисходящему компоненту, не может быть автоматически завершена, для этого связь должна быть сначала удалена из использующих ее Соединений. Кроме того, у предназначенной к автоматическому прерыванию связи при ее добавлении к Соединению статус "auto-termination" отключается.

6.2 Вкладка SCHEDULING

"SCHEDULING" – вторая вкладка диалогового окна настроек Процессора (Рис.6.2.).

Вкладка содержит несколько элементов конфигурации:

- *Scheduling Strategy*
- *Concurrent Tasks*
- *Run Schedule*
- *Execution*
- *Run Duration*

6.2.1 Scheduling Strategy

В поле стратегии планирования "Scheduling Strategy" есть три возможных варианта планирования компонентов:

- *Timer driven* – режим по умолчанию – запуск Процессора осуществляется с регулярным интервалом, определенным опцией "Run Schedule";
- *Event driven* – в данном режиме Процессор инициируется с выполнением события, когда FlowFiles входят в Соединения, относящиеся к Процессору. Режим в настоящее время считается экспериментальным и не поддерживается всеми Процессорами. В данном режиме опция "Run Schedule" недоступна для конфигурации, так как Процессор запускается не по периодам, а в результате выполнения события. Кроме того, это единственный режим, для которого параметр "Concurrent Tasks" может быть установлен равным *0*, так как количество потоков ограничено только размером Event-Driven Thread Pool, настроенным администратором;
- *CRON driven* – запуск Процессора осуществляется периодически, подобно режиму *Timer driven*. Однако *CRON driven* обеспечивает значительно большую гибкость за счет увеличения сложности конфигурации, представляющую собой строку из шести обязательных полей и одного опционального, разделенных пробелом.

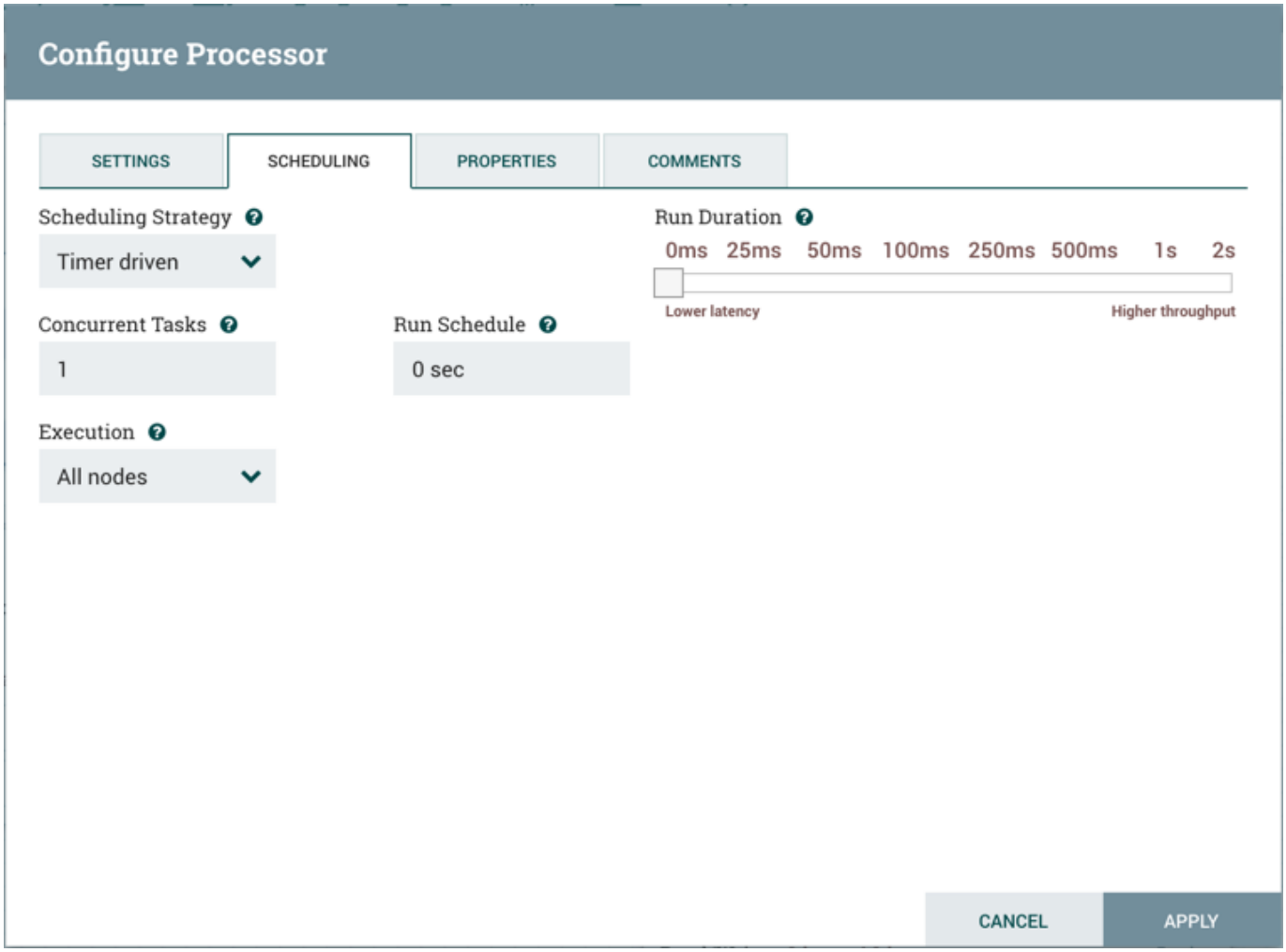


Рис.6.2.: Вкладка SCHEDULING

Таблица6.1.: Конфигурация стратегии планирования CRON driven

Поле	Допустимые значения
Seconds	0-59
Minutes	0-59
Hours	0-23
Day of Month	1-31
Month	1-12 или JAN-DEC
Day of Week	1-7 или SUN-SAT
Year (опционально)	Пусто или 1970-2099

Значения задаются одним из следующих способов:

- *Number* – одно или несколько допустимых значений, разделенных запятыми;
- *Range* – диапазон значений в виде <number>-<number>;
- *Increment* – инкремент с использованием синтаксиса <start value>/<increment>. Например, в поле “Minutes” значение *0/15* обозначает последовательность минут *0*, *15*, *30* и *45*.

Кроме того, могут быть использованы специальные символы:

- Символ * – означает, что все допустимые значения действительны;
- Символ ? – означает, что может быть задано не характерное значение (допустипо в полях “Day of Month” и “Day of Week”);
- Символ L – можно добавить *L* к одному из значений дня недели, чтобы указать последнее вхождение этого дня в месяце. Например, *1L* обозначает последнее воскресенье месяца.

Примеры:

- Строка `0 0 13 * * ?` указывает, что необходимо запланировать запуск Процессора в 13:00 каждый день;
- Строка `0 20 14 ? * MON-FRI` указывает, что необходимо запланировать запуск Процессора в 14:20 с понедельника по пятницу;
- Строка `0 15 10 ? * 6L 2011-2017` указывает, что необходимо запланировать запуск Процессора в 10:15 в последнюю пятницу каждого месяца в период с 2011 по 2017 год.

Дополнительную информацию с примерами можно найти в документации Quartz по ссылке Chron Trigger Tutorial.

6.2.2 Concurrent Tasks

Параметр конфигурации “Concurrent Tasks” – параллельные задачи – определяет количество потоков, используемых Процессором, то есть количество одновременно обрабатываемых FlowFiles. Увеличение значения, как правило, позволяет Процессору обрабатывать больше данных за тот же промежуток времени. Однако это достигается засчет использования системных ресурсов, которые в таком случае не могут использоваться другими Процессорами. Параметр по существу контролирует, сколько ресурсов системы должно быть выделено для конкретного Процессора.

Поле “Concurrent Tasks” доступно для большинства Процессоров. Однако существуют некоторые типы Процессоров, которые можно запланировать только с одной параллельной задачей.

6.2.3 Run Schedule

Параметр “Run Schedule” определяет расписание запуска Процессора. Допустимые значения для поля зависят от выбранной стратегии планирования “Scheduling Strategy”. При стратегии *Event driven* поле “Run Schedule” недоступно. При стратегии *Timer driven* значение представляет собой единицу времени. Например, *1 second* или *5 mins*. Значение по умолчанию *0 sec* означает, что Процессор должен работать как можно чаще при наличии данных для обработки. Условие верно для любой продолжительности времени со значением *0* (независимо от единицы времени, то есть *0 sec*, *0 mins*, *0 days*). Объяснение значений, применимых к стратегии *CRON driven*, приведено в описании самой стратегии.

6.2.4 Execution

Параметр “Execution” используется для определения узла, на котором запланирован запуск Процессора. Выбор значения *All Nodes* приводит к планированию запуска Процессора на каждом узле кластера. Значение *Primary Node* приводит к тому, что запуск Процессора планируется только на первичном узле. Настроенные на *Primary Node* Процессоры помечаются значком с буквой “P” рядом с пиктограммой самого Процессора (Рис.6.3.).

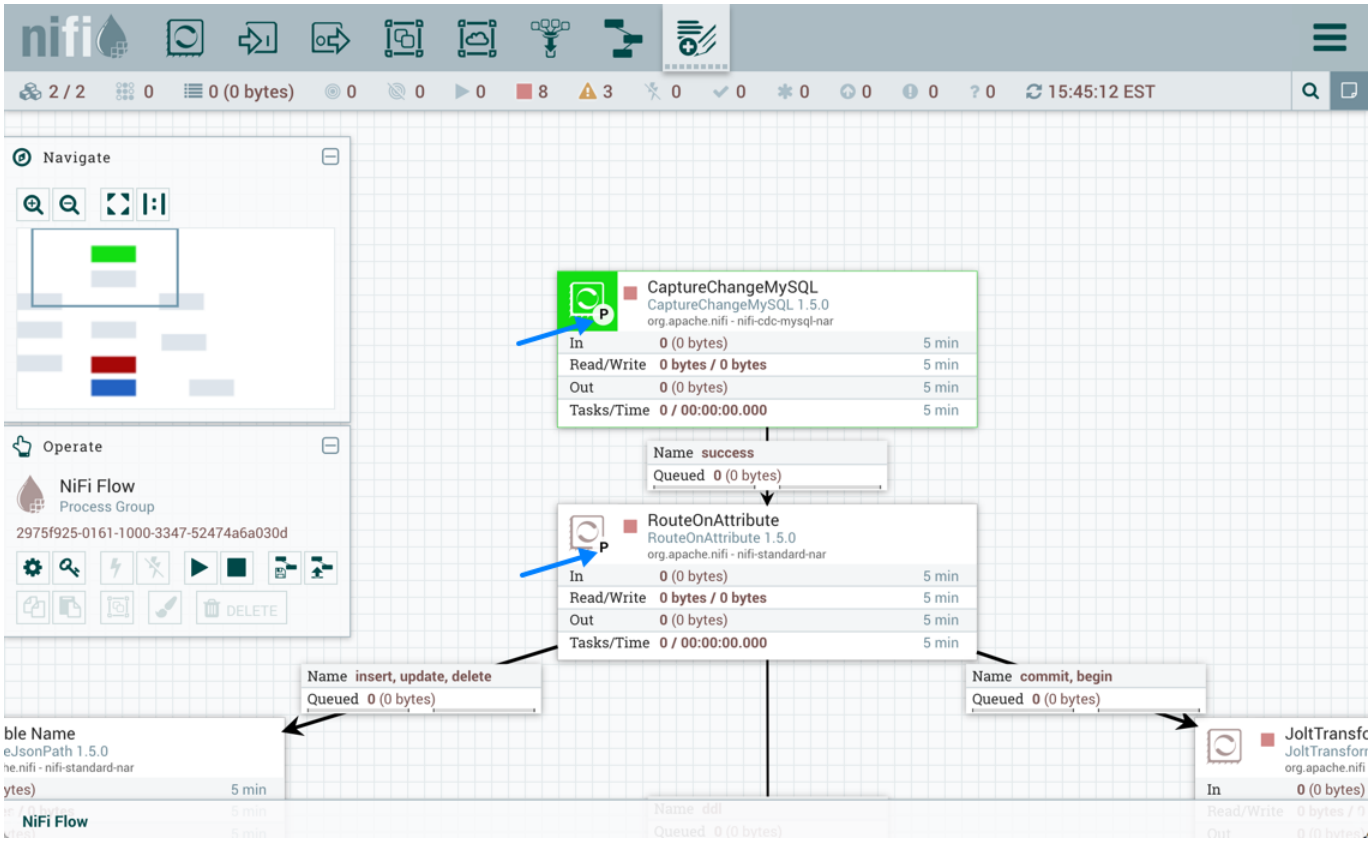


Рис.6.3.: Процессоры, настроенные на Primary Node

Для быстрого определения настроенных на первичный узел Процессоров значок “P” также отображается на вкладке “Processors” на странице “Summary” (Рис.6.4.).

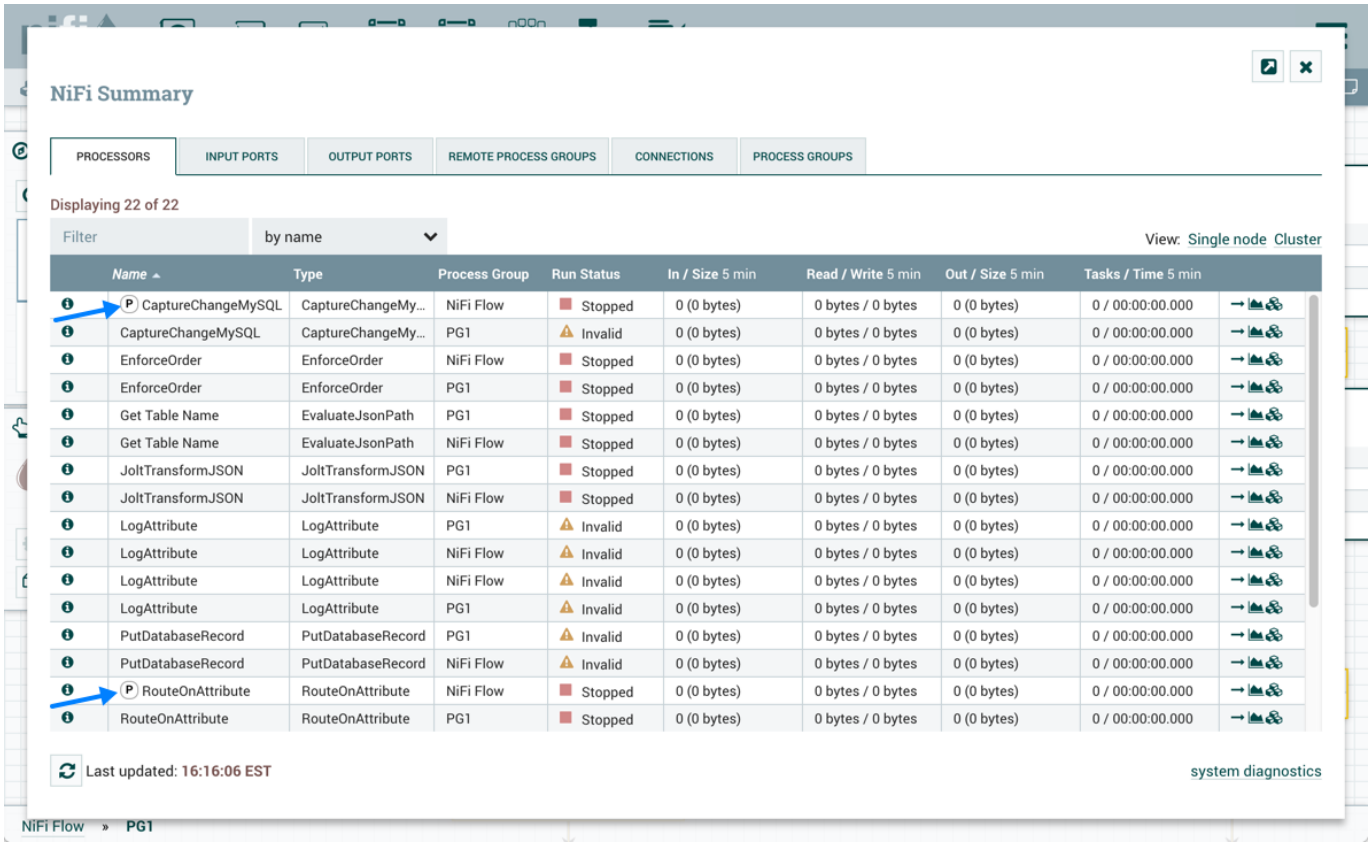


Рис.6.4.: Процессоры, настроенные на Primary Node

6.2.5 Run Duration

Правая часть вкладки “SCHEDULING” содержит ползунок для управления параметром “Run Duration”, определяющий длительность повторных запусков Процессора. Левая часть ползунка помечена как нижняя грань латентности “Lower latency”, правая – как наиболее высокая пропускная способность “Higher throughput”.

При завершении работы Процессор обновляет репозиторий, чтобы передать FlowFiles следующему Соединению. Обновление хранилища является ресурсоемким процессом, поэтому, чем больше работы может быть выполнено перед обновлением репозитория, тем больше работы Процессор может обработать (более высокая пропускная способность). Однако, это означает, что следующий Процессор не может начать обработку задействованных потоков, пока предыдущий процесс не обновит репозиторий, что приводит к увеличению латентности (время, необходимое для обработки FlowFile от начала до конца, увеличивается). В результате, ползунок параметра “Run Duration” предоставляет спектр, из которого DFM выбирает более низкую задержку или более высокую пропускную способность.

6.3 Вкладка PROPERTIES

Вкладка свойств “PROPERTIES” предоставляет механизм для настройки поведения Процессора. Свойства не задаются по умолчанию и каждому типу Процессора определяются характерные ему оцпии, имеющие смысл для конкретного использования. Далее приведены свойства для Процессора *RouteOnAttribute* (Рис.6.5.).

Данный Процессор имеет одно свойство по умолчанию – “Routing Strategy”, заданное на “Route to Property name”. Рядом с наименованием свойства находится символ знака вопроса, и как и в других местах интерфейса символ обозначает информационную справку для пользователя, при наведении курсора на которую в данном случае выдается дополнительная информация о свойстве и установленном значении по умолчанию, а также его исторические значения.

Клик по значению свойства позволяет DFM изменить его. В зависимости от допустимых значений для конкретного свойства предоставляется либо раскрывающийся список, либо открывается текстовое поле для ввода (Рис.6.6.).

В правом верхнем углу вкладки расположена кнопка добавления свойства “New Property”, при нажатии на которую открывается диалоговое окно для ввода имени и значения нового свойства. Не всеми Процессорами допускаются пользовательские свойства User-Defined, и в случае их назначения Процессор становится недействителен (Рис.6.7.).

После добавления свойства User-Defined в правой части его строки появляется значок удаления, при нажатии на который свойство удаляется из Процессора.

Некоторые Процессоры, например, *UpdateAttribute*, имеют встроенный пользовательский интерфейс. Для перехода к нему необходимо нажать кнопку “Advanced”, которая появляется в нижней части окна настройки у подобных Процессоров.

Так же некоторые Процессоры имеют свойства, ссылающиеся на другие компоненты, например, на Controller Services, которые также требуют настройки. Например, Процессор *GetHTTP* имеет свойство *SSLContextService*, которое ссылается на контроллер *StandardSSLContextService*. В случае, когда DFM необходимо настроить свойство, но при этом еще не создан и не настроен контроллер, у DFM есть возможность сделать это сразу на месте, как показано далее на рисунке (Рис.6.8.).

6.4 Вкладка COMMENTS

Последней вкладкой диалогового окна конфигурации Процессора является вкладка “COMMENTS”, предоставляющая пользователям область для добавления комментариев к компоненту. Использование вкладки необязательно (Рис.6.9.).

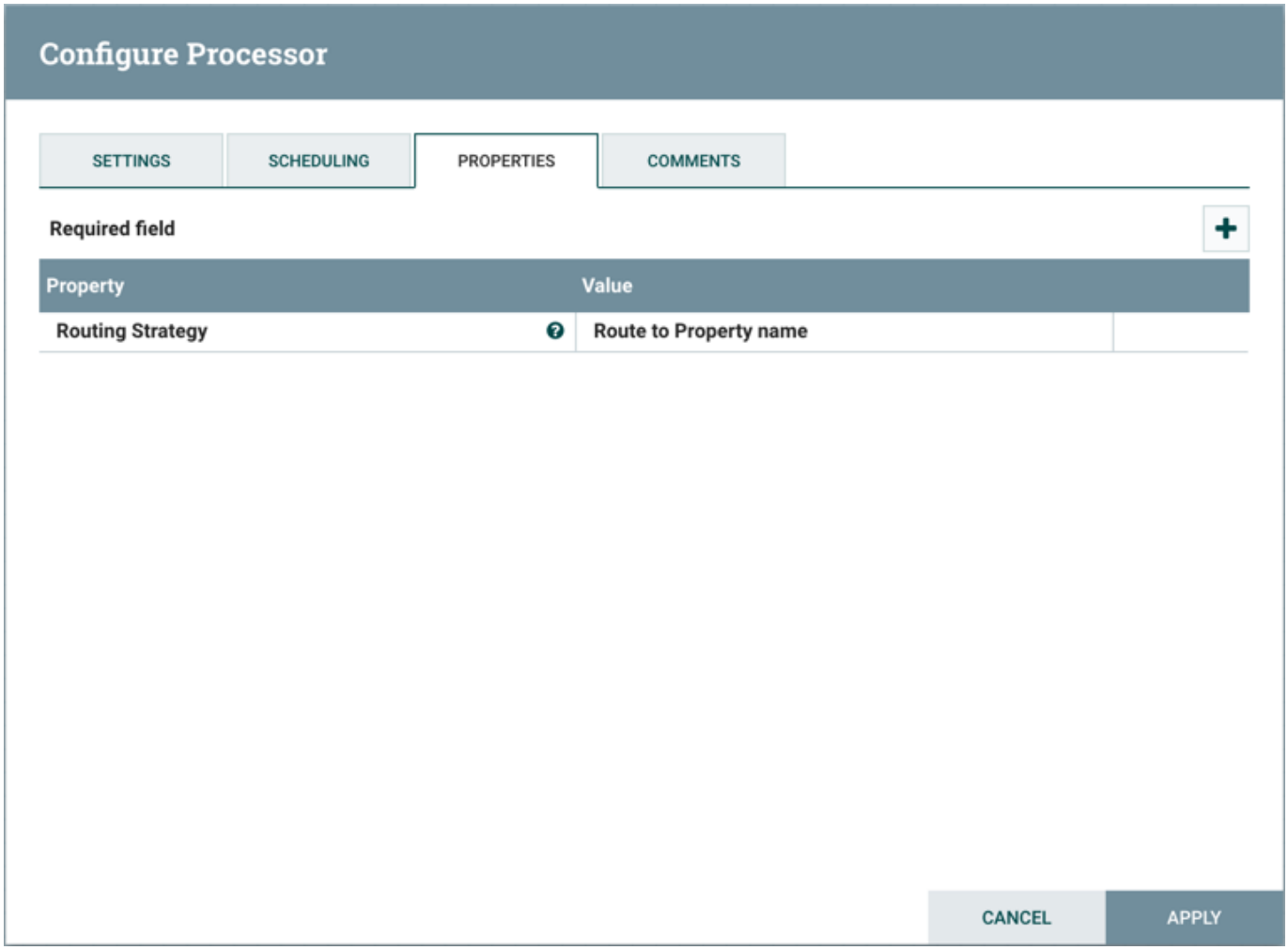


Рис.6.5.: Вкладка PROPERTIES

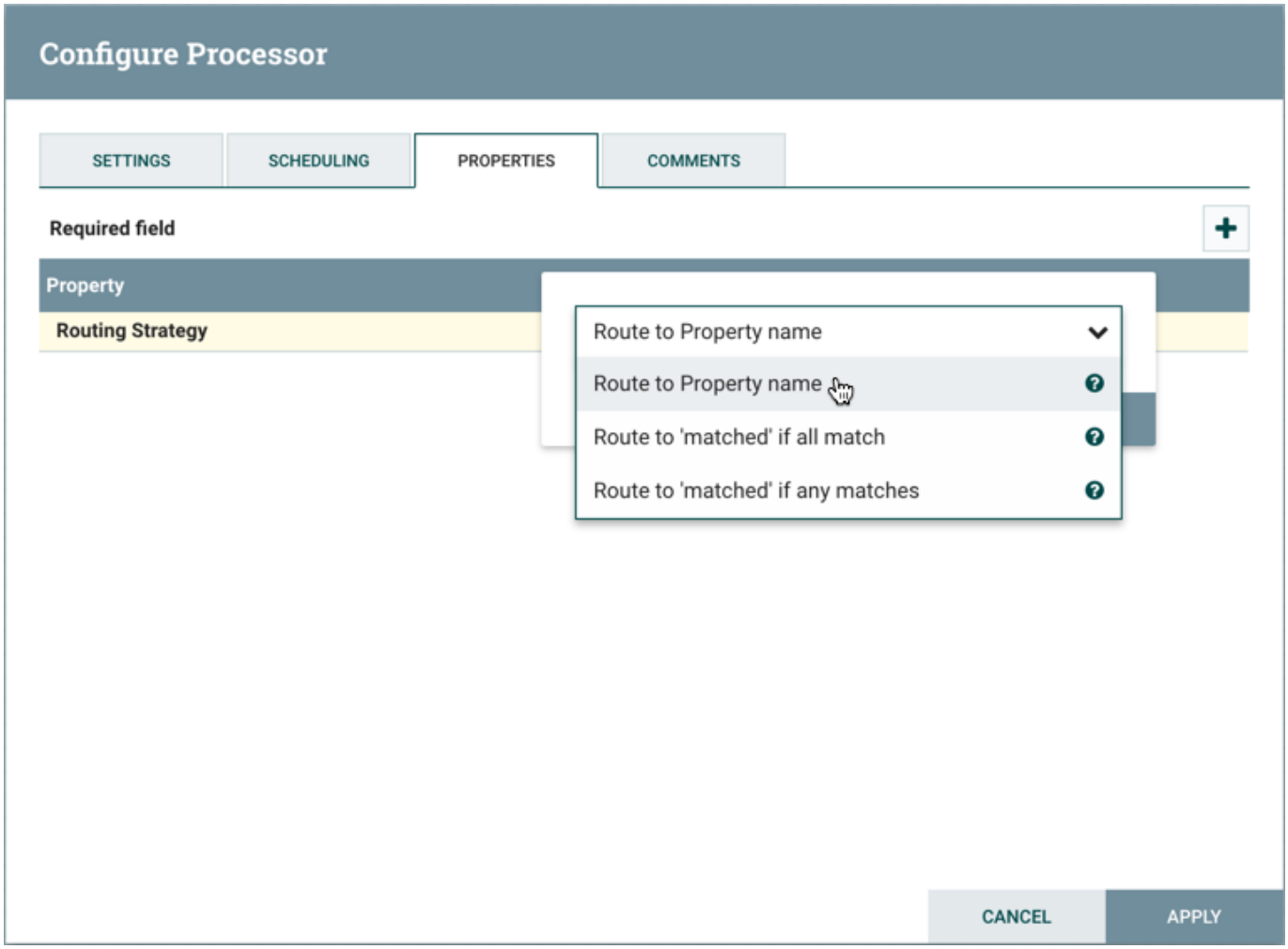


Рис.6.6.: Выбор значения свойства

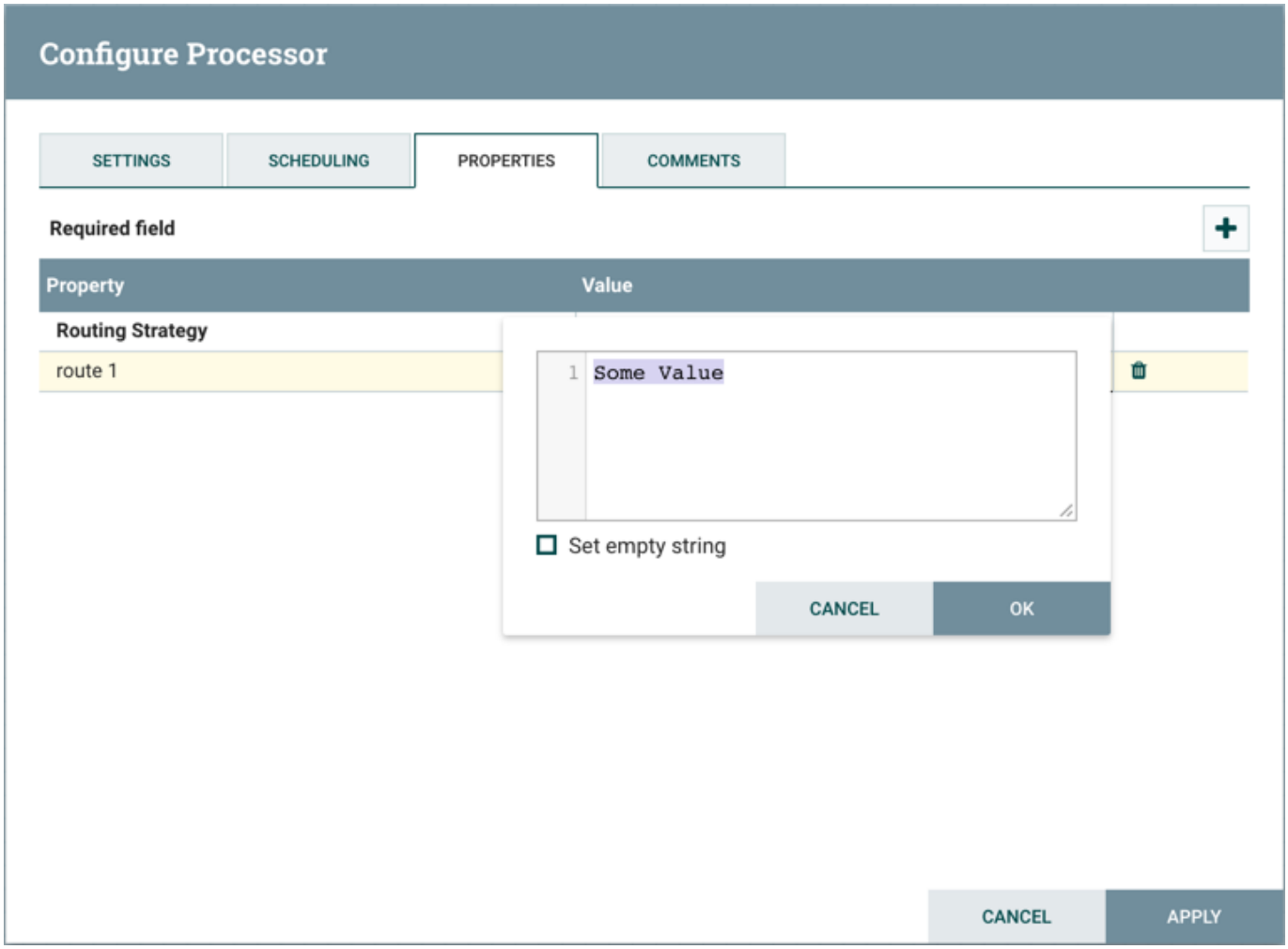


Рис.6.7.: Добавление свойства

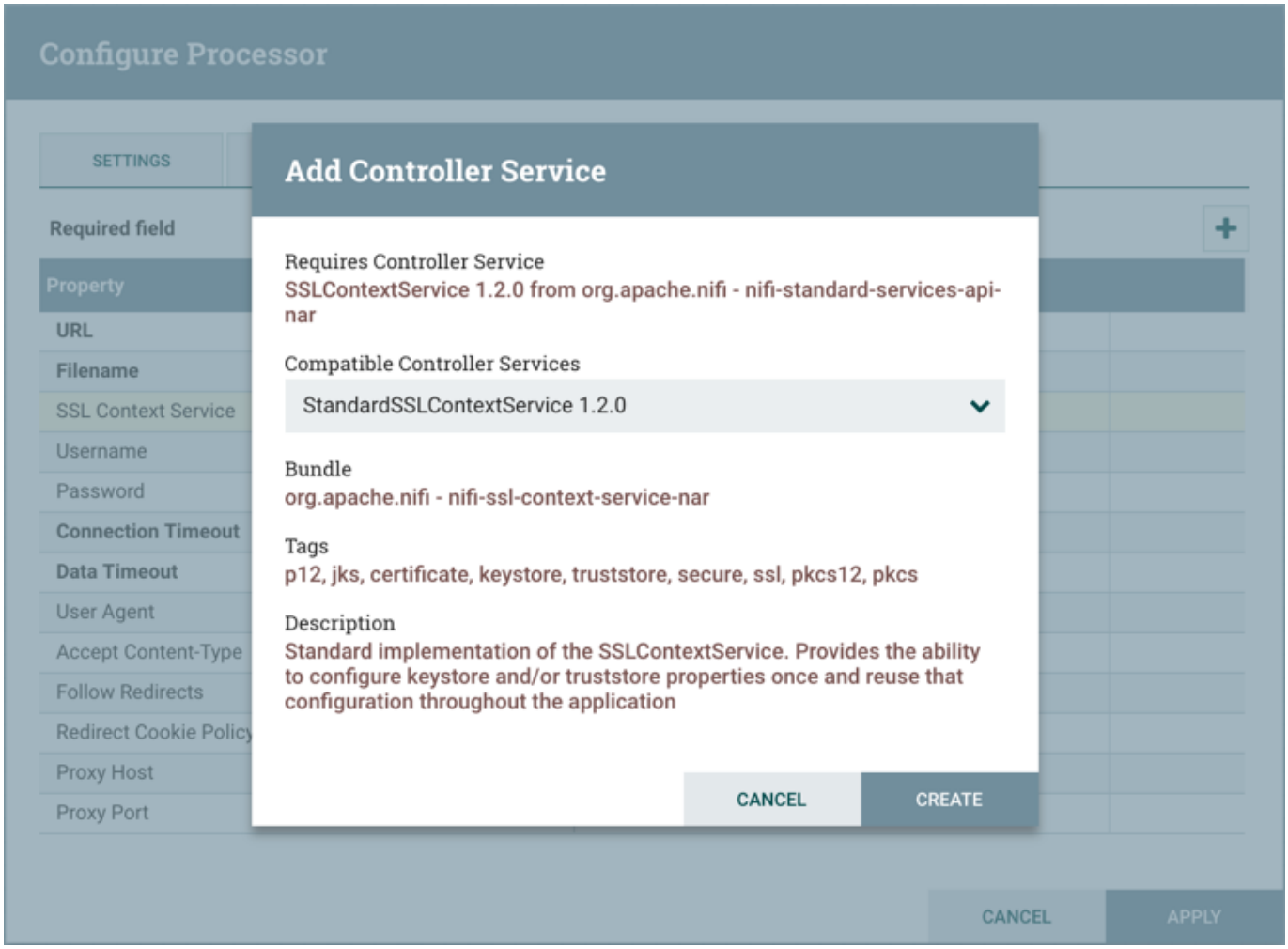


Рис.6.8.: Настройка контроллера через свойство Процессора

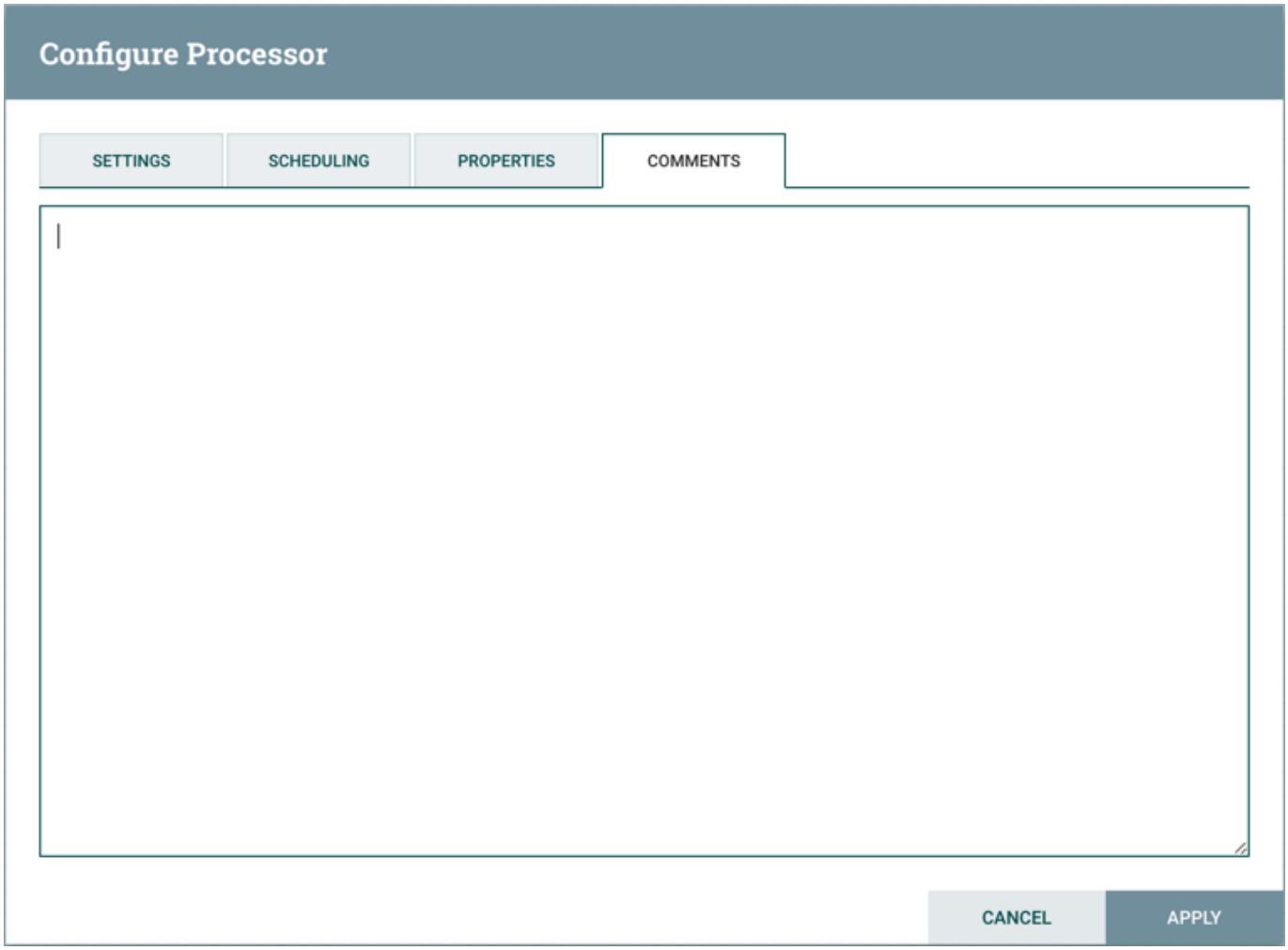


Рис.6.9.: Вкладка COMMENTS

Глава 7

Пользовательские свойства с использованием Expression Language

Expression Language (язык выражений) **NiFi** можно использовать для отображения значений атрибутов FlowFile, сравнения их с другими значениями и управления ими при создании и настройке потоков данных. Сведения о языке выражений по ссылке Expression Language Guide.

В дополнение к использованию атрибутов FlowFile, системных свойств и свойств среды с помощью Express Language также можно определить пользовательские свойства, что обеспечивает большую гибкость в управлении и обработке потоков данных. Также возможно создание пользовательских свойств для соединения, сервера и сервиса с целью упрощения настройки потока данных.

Свойства **NiFi** имеют приоритет разрешения, о котором следует знать при создании пользовательских свойств:

- Атрибуты процессора;
- Свойства FlowFile;
- Атрибуты FlowFile;
- Из реестра переменных:
  - Пользовательские свойства;
  - Системные свойства;
  - Переменные среды операционной системы.

Important: При создании пользовательских свойств необходимо убедиться, что каждое такое свойство содержит отдельное значение с целью предотвращения его переопределения существующими свойствами среды, системными свойствами или атрибутами FlowFile

Есть два способа использования и управления пользовательскими свойствами:

- В интерфейсе NiFi через окно переменных “Variables”;
- По ссылке *nifi.properties*.

7.1 Окно переменных Variables

Переменные могут быть созданы и настроены в пользовательском интерфейсе **NiFi** в любом поле, поддерживающем Expression Language. **NiFi** автоматически подбирает новые или измененные переменные, созданные в UI.

Для перехода к диалоговому окну “Variables” необходимо кликнуть правой кнопкой мыши по пустому пространству рабочей области приложения и выбрать пункт контекстного меню “Variables” (Рис.7.1.).

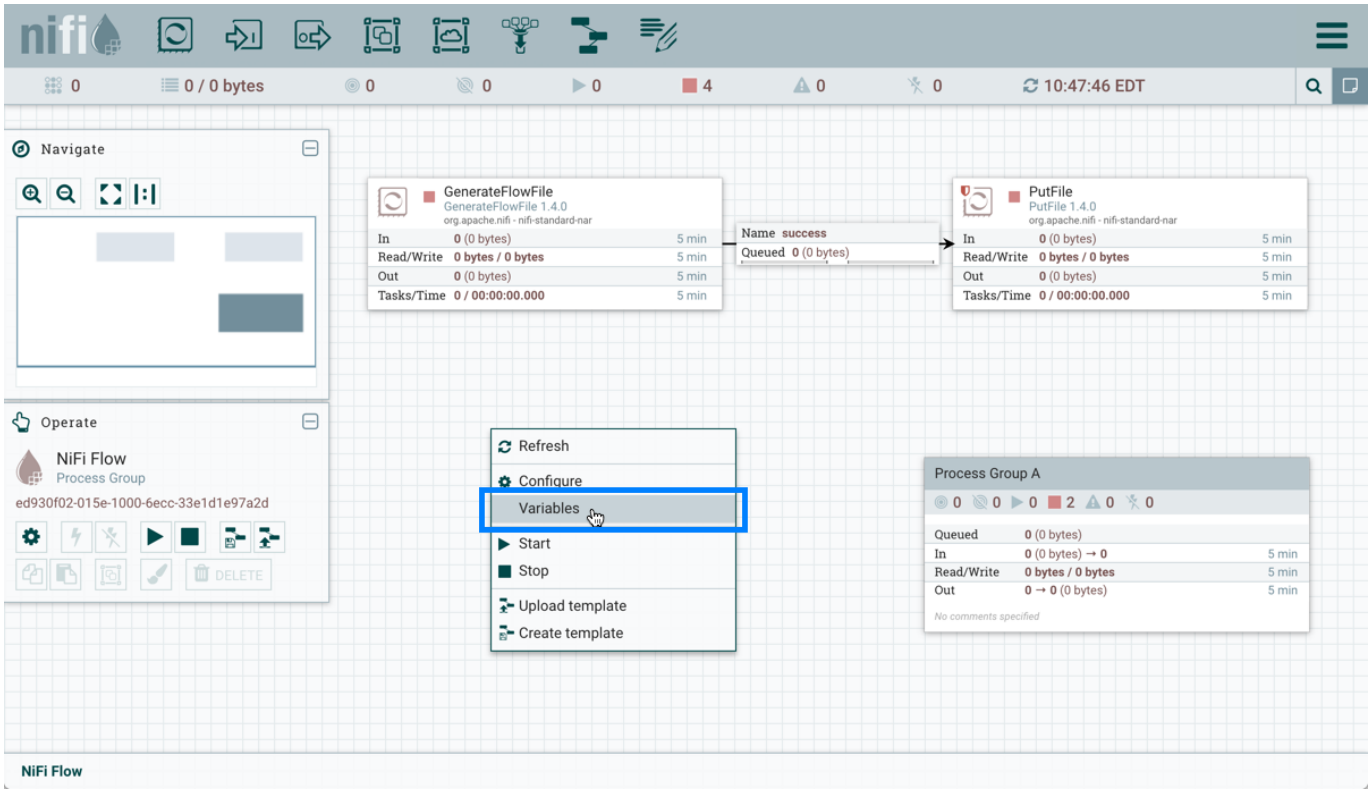


Рис.7.1.: Переход к “Variables” с рабочей области NiFi

При выборе группы процессов переход к “Variables” также доступен из контекстного меню (Рис.7.2.).

Диалоговое окно создания и настроек пользовательских свойств “Variables” представлено на рисунке (Рис.7.3.).

7.1.1 Создание переменной

Для создания новой переменной необходимо в окне “Variables” нажать кнопку с символом “+” и в открывшейся форме ввести наименование (Рис.7.4.).

Затем нажать кнопку “OK” и в новом окне ввести значение переменной (Рис.7.5.).

После чего в экранной форме “Variables” нажать кнопку “Apply”, результатом действия которой является иформационное окно по обновлению/созданию переменной (Рис.7.6.).

При этом запускается процесс обновления переменной (идентификация затронутых компонентов, остановка затронутых процессоров и т.д.). Например, в разделе “Referencing Processors” теперь отображается процессор “PutFile-Root”. Выбор имени процессора в списке приводит к переходу к указанному процессору на

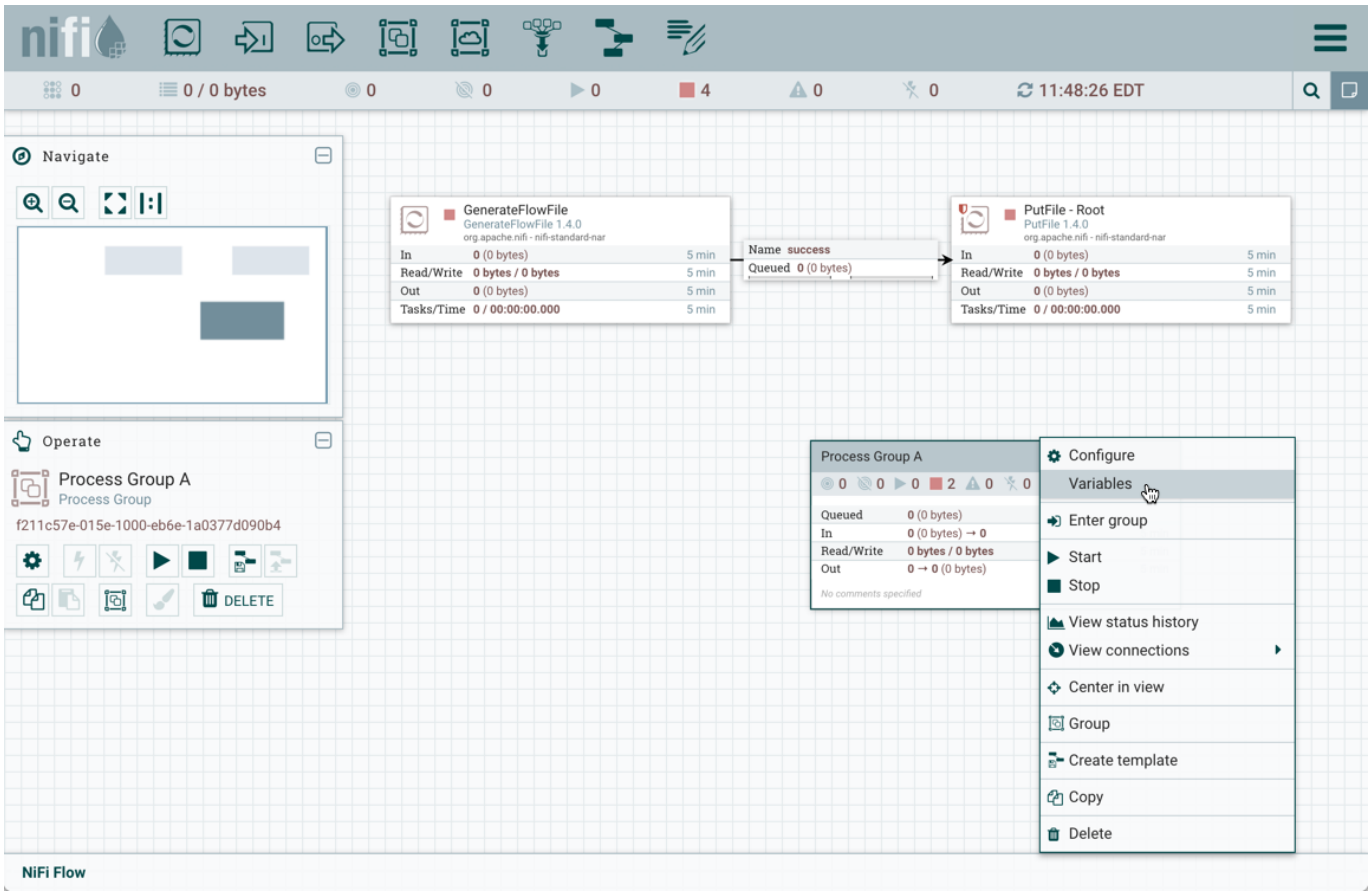


Рис.7.2.: Переход к “Variables” через группу процессов

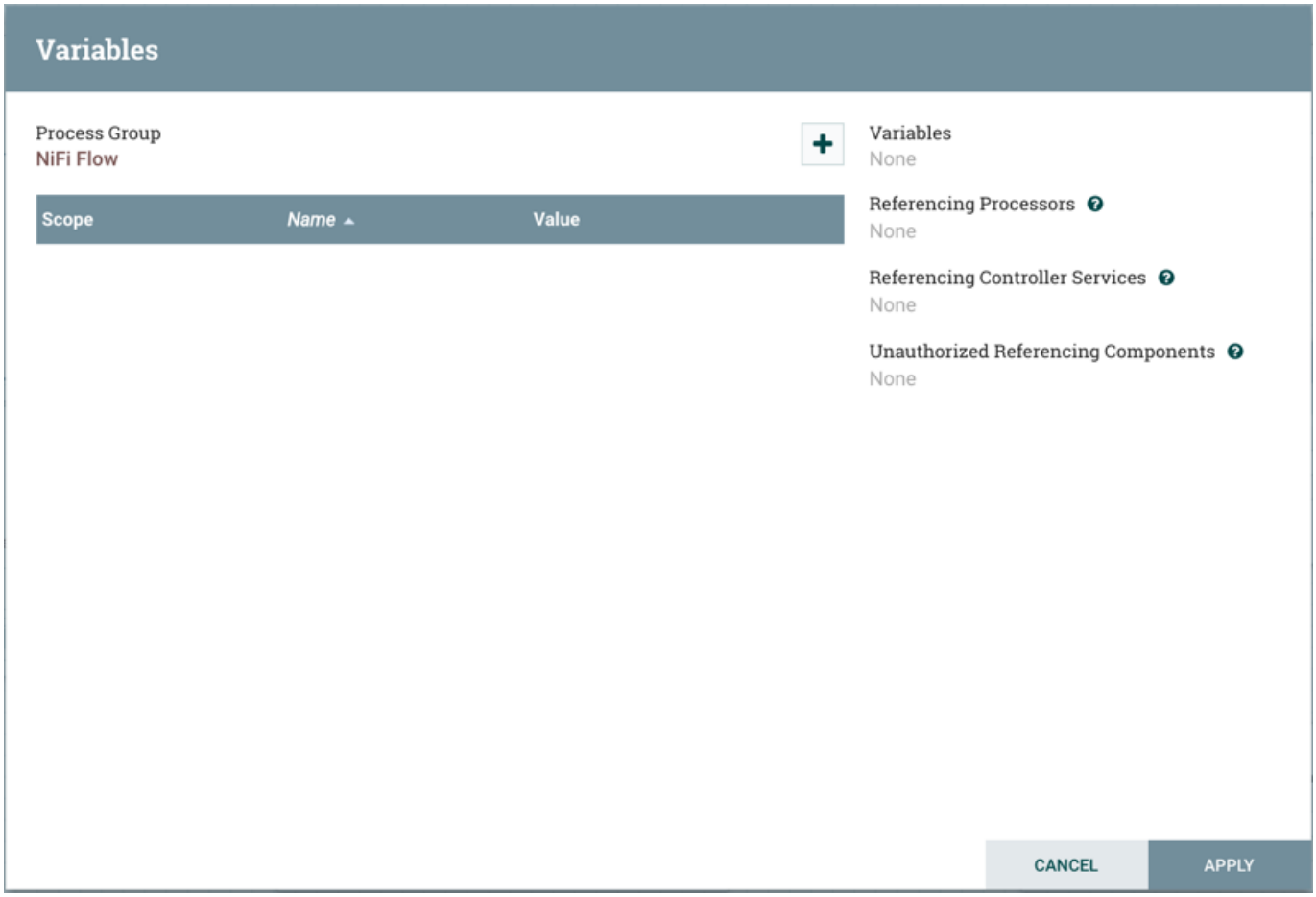


Рис.7.3.: Диалоговое окно “Variables”

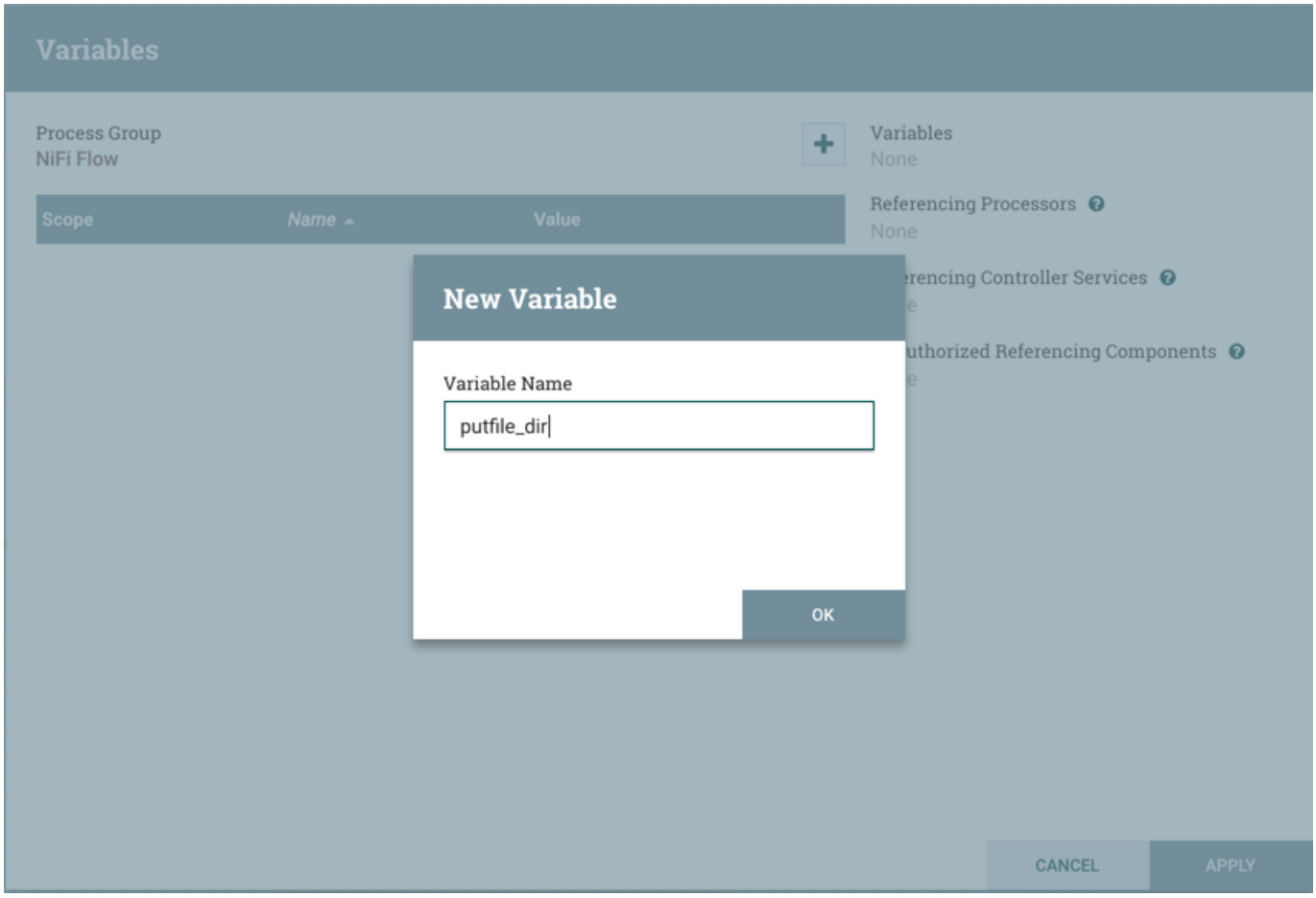


Рис. 7.4.: Наименование новой переменной

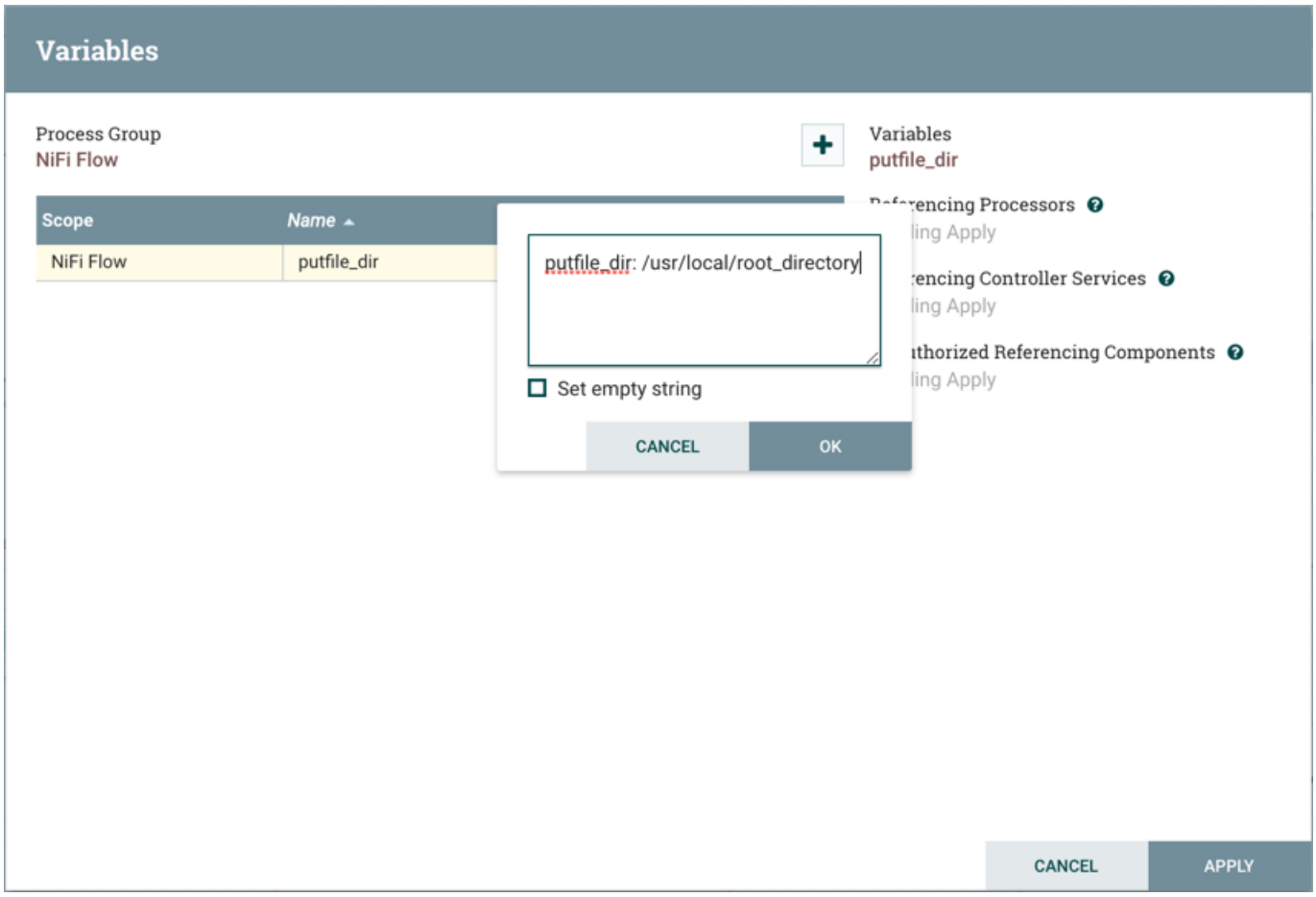


Рис.7.5.: Значение новой переменной

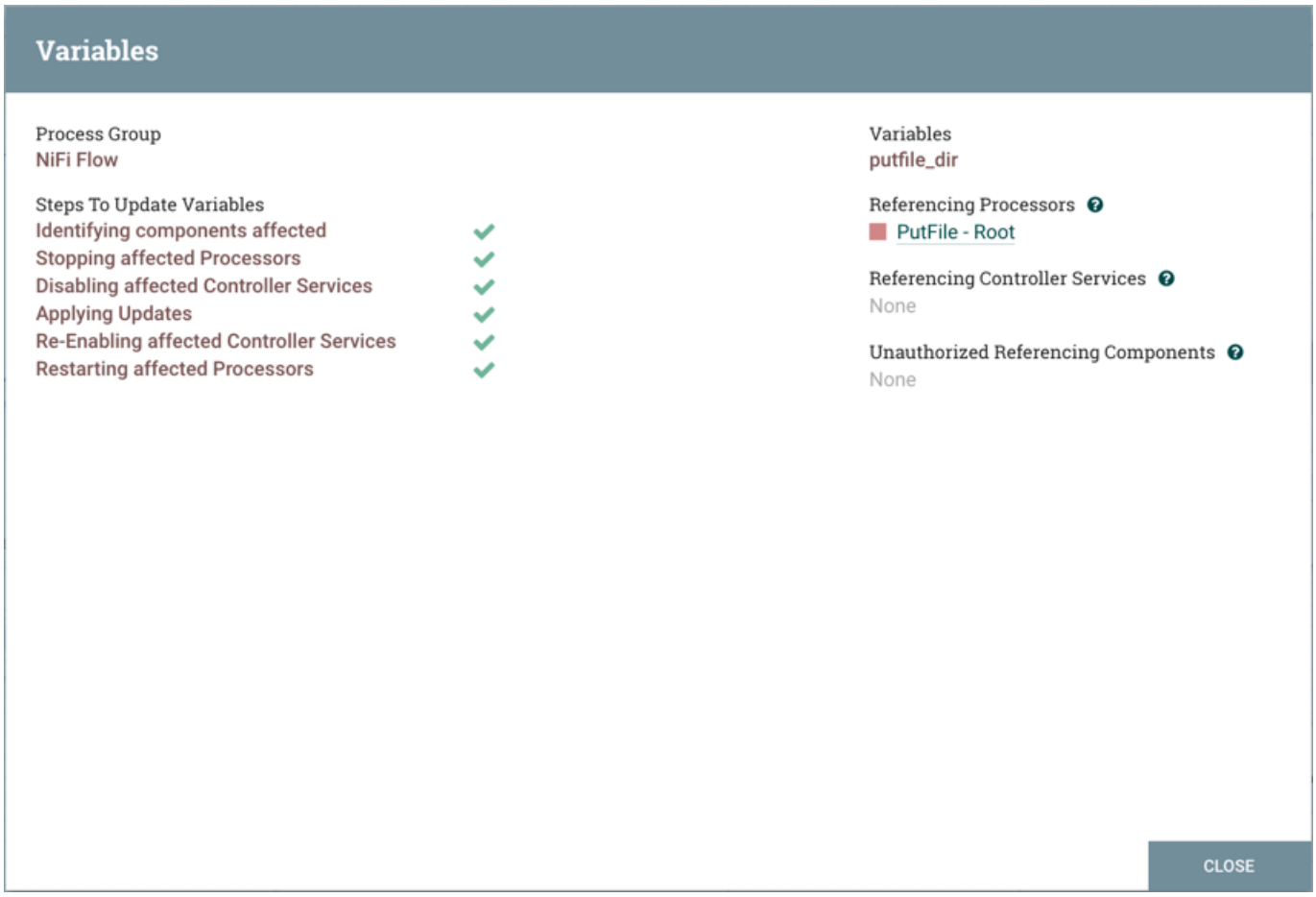


Рис.7.6.: Обновление переменной

рабочей области. А при просмотре свойств процессора свойство *Directory* ссылается на созданную переменную *${putfile_dir}* (Рис.7.7.).

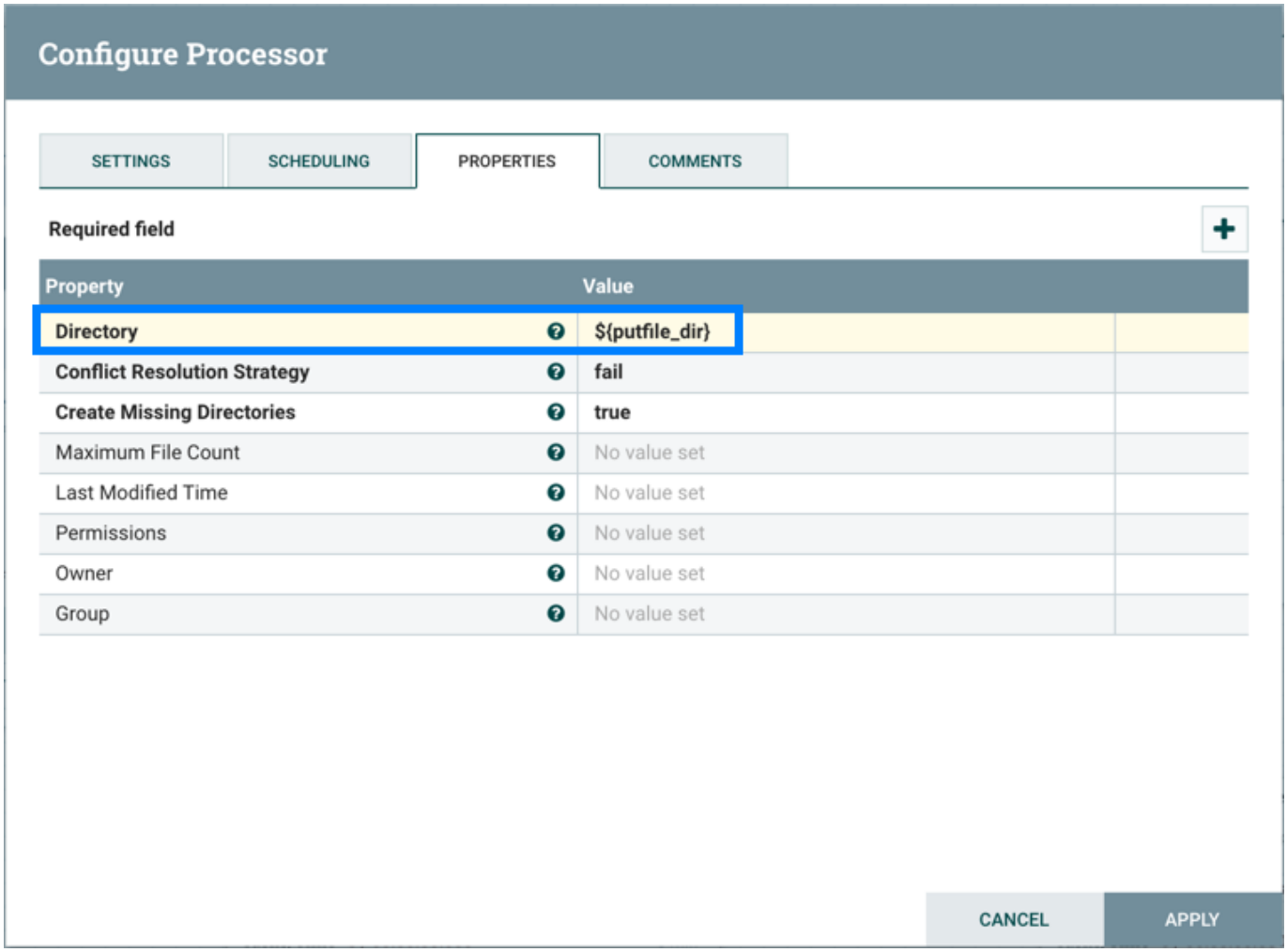


Рис.7.7.: Переменная в свойствах процессора

7.1.2 Область действия переменной

Область действия переменных определяется группой процессов, в которой они заданы, и доступны любому Процессору, определенному на данном уровне и ниже (то есть любому наследованному Процессору).

При этом переменные в наследованной группе переопределяют значения в родительской группе. В частности, если переменная задана в группе *root*, а также получает иное значение внутри группы процессов, то в таком случае компоненты внутри группы процессов используют значение, определенное непосредственно в самой группе процессов.

Например, переменная *putfile_dir* существует в группе процессов *root*, и в то же время создается другая переменная *putfile_dir* в группе процессов *A*. В таком случае, если один из компонентов в группе процессов *A* ссылается на переменную *putfile_dir*, то указываются обе переменные, но *putfile_dir* из группы *root* при этом перечеркнута, так как она переопределена (Рис.7.8.).

Значение переменной может быть изменено только в группе процессов, в которой она создана (данная группа указывается в верхней части окна “Variables”). Для изменения переменной, определенной в другой группе

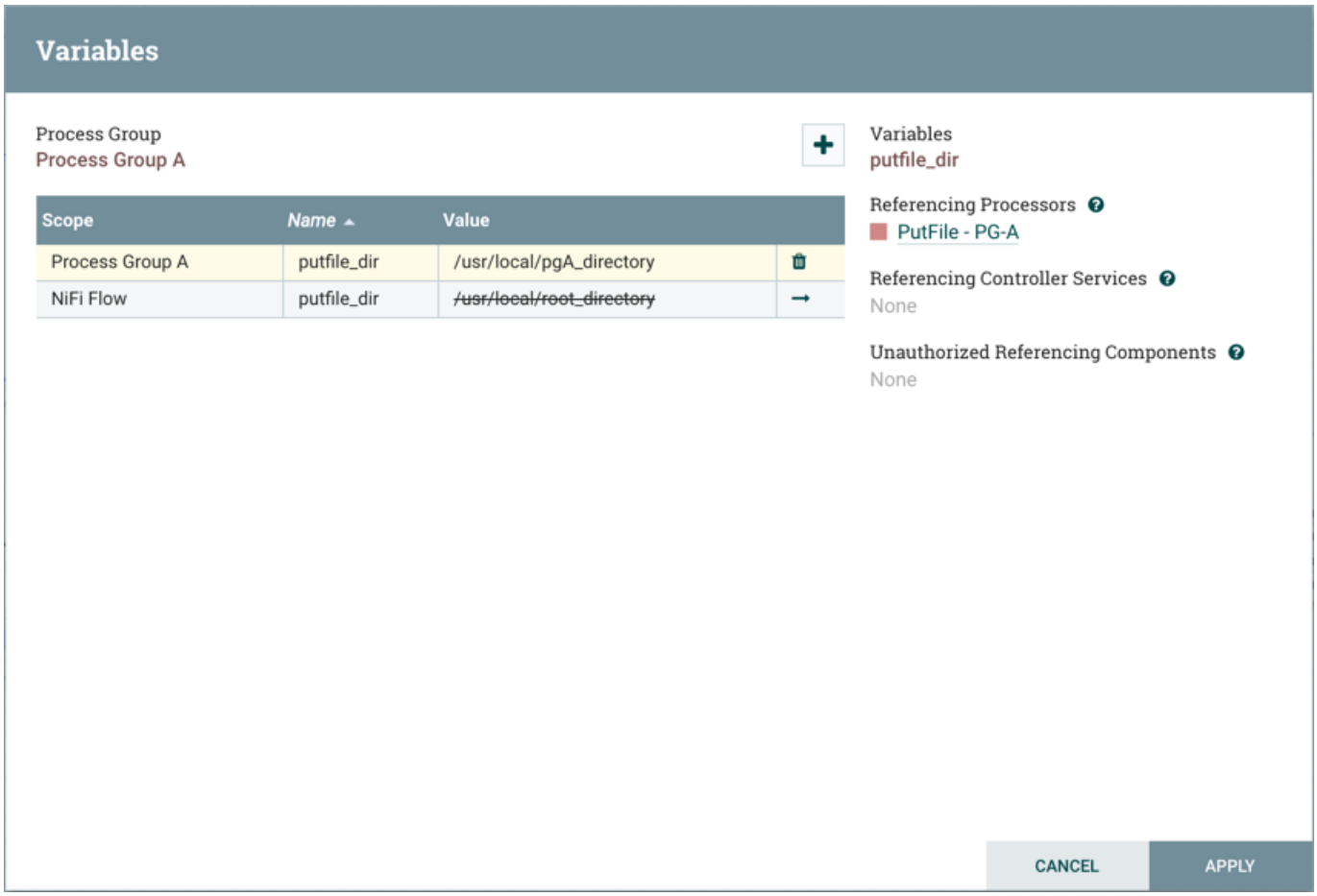


Рис.7.8.: Переопределение переменной

процессов, необходимо выбрать значок стрелки в строке интересующей переменной (Рис.7.9.).

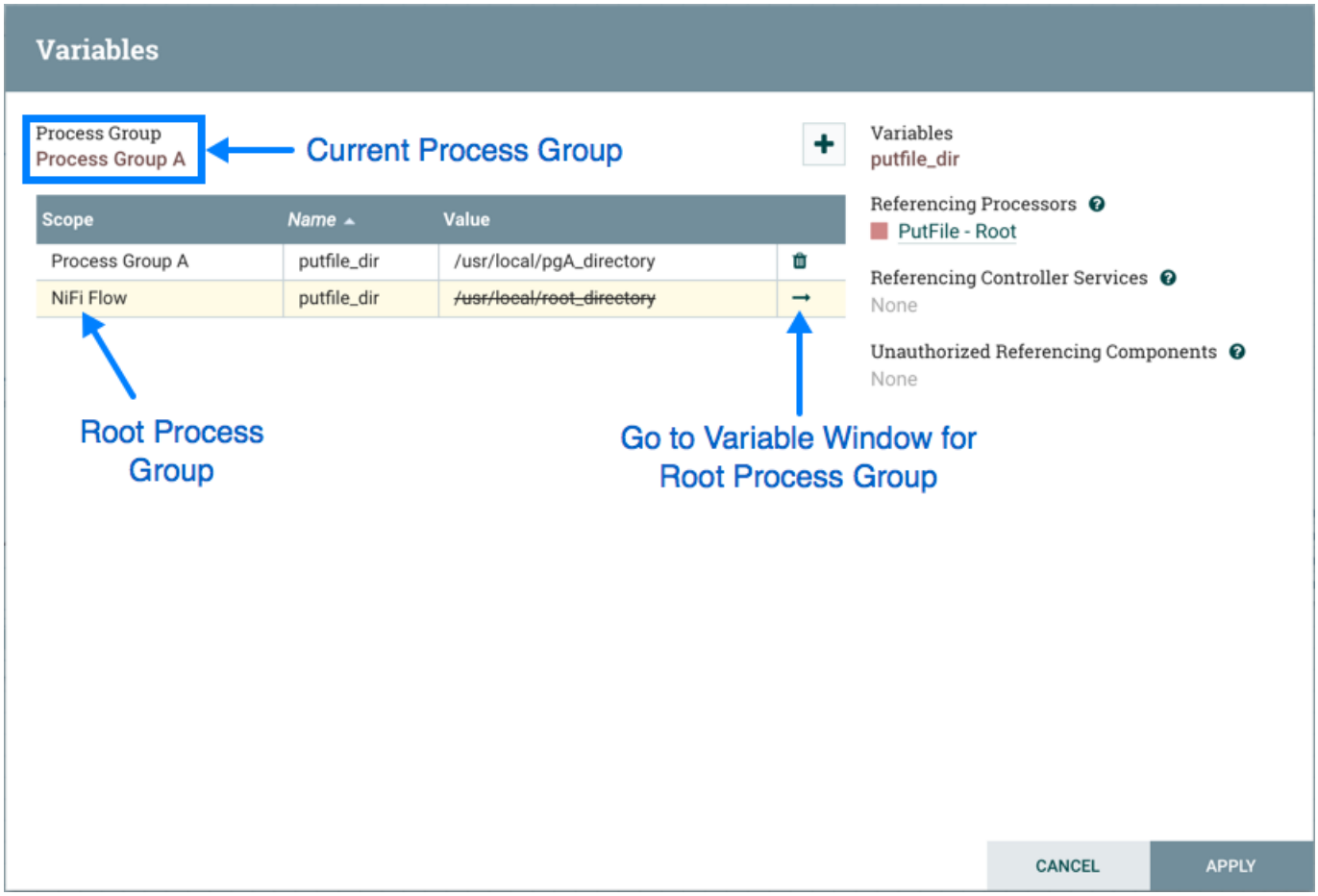


Рис.7.9.: Изменение переменной

При этом происходит переход к экранной форме “Variables” группы процессов, создавшей переменную (Рис.7.10.).

7.1.3 Разрешения переменных

Разрешения переменных основаны исключительно на привилегиях, настроенных для соответствующей группы процессов. Например, если у пользователя нет доступа к просмотру группы процессов (“View a process group”), окно “Variables” для данной группы процессов не может быть открыто (Рис.7.11.).

При наличии у пользователя прав доступа к просмотру группы процессов, но при этом отсутствии доступа к изменению настроек (“Modify the process group”), переменные так же можно только просматривать, но не изменять.

Сведения об управлении привилегиями компонентов приведены в разделе Политики доступа документа Руководство администратора по работе с сервисом Nifi.

7.1.4 Controller Services

В экранной форме “Variables” также отображаются ссылки на контроллеры (Рис.7.12.).

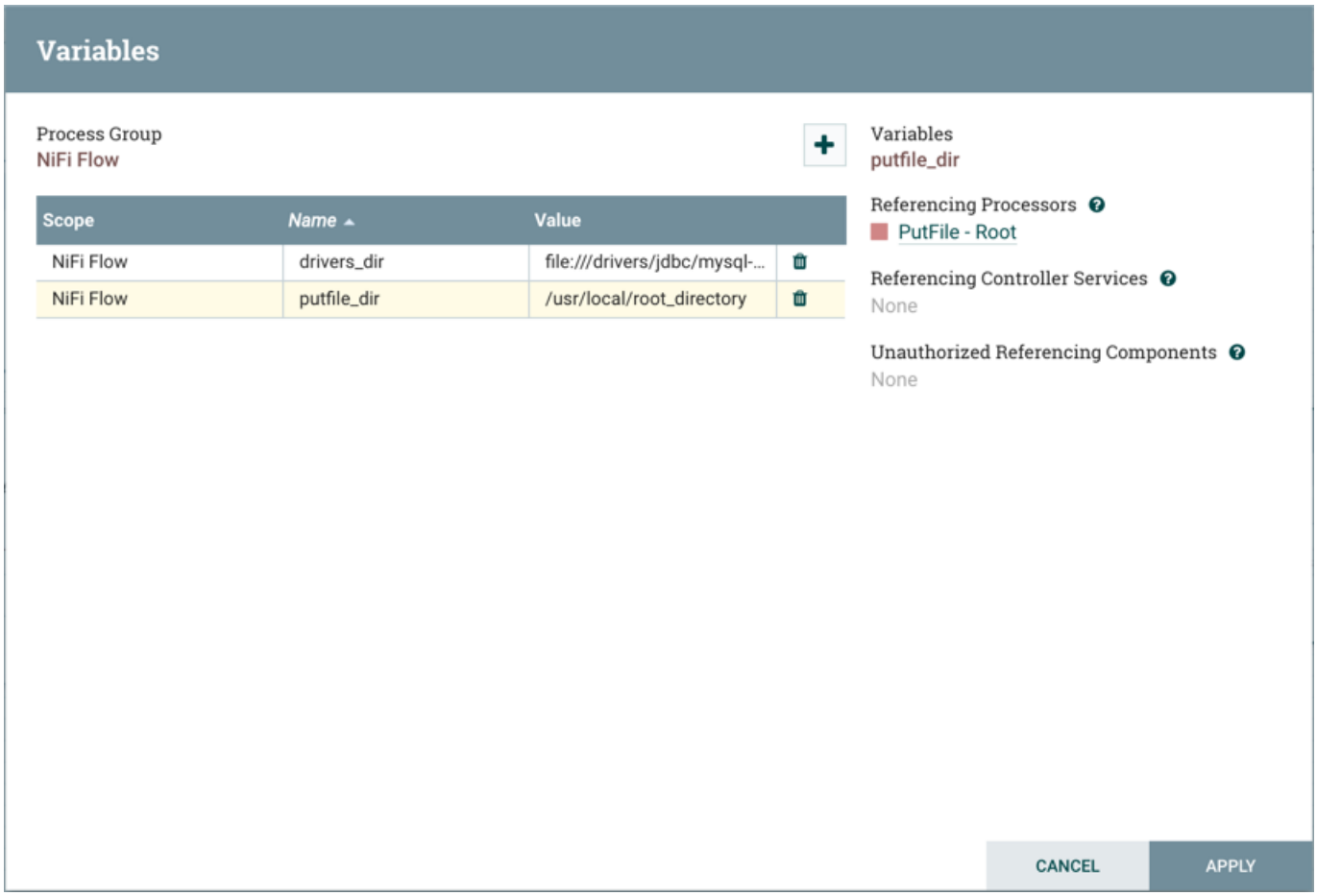


Рис.7.10.: Переход к группе процессов, создавшей переменную

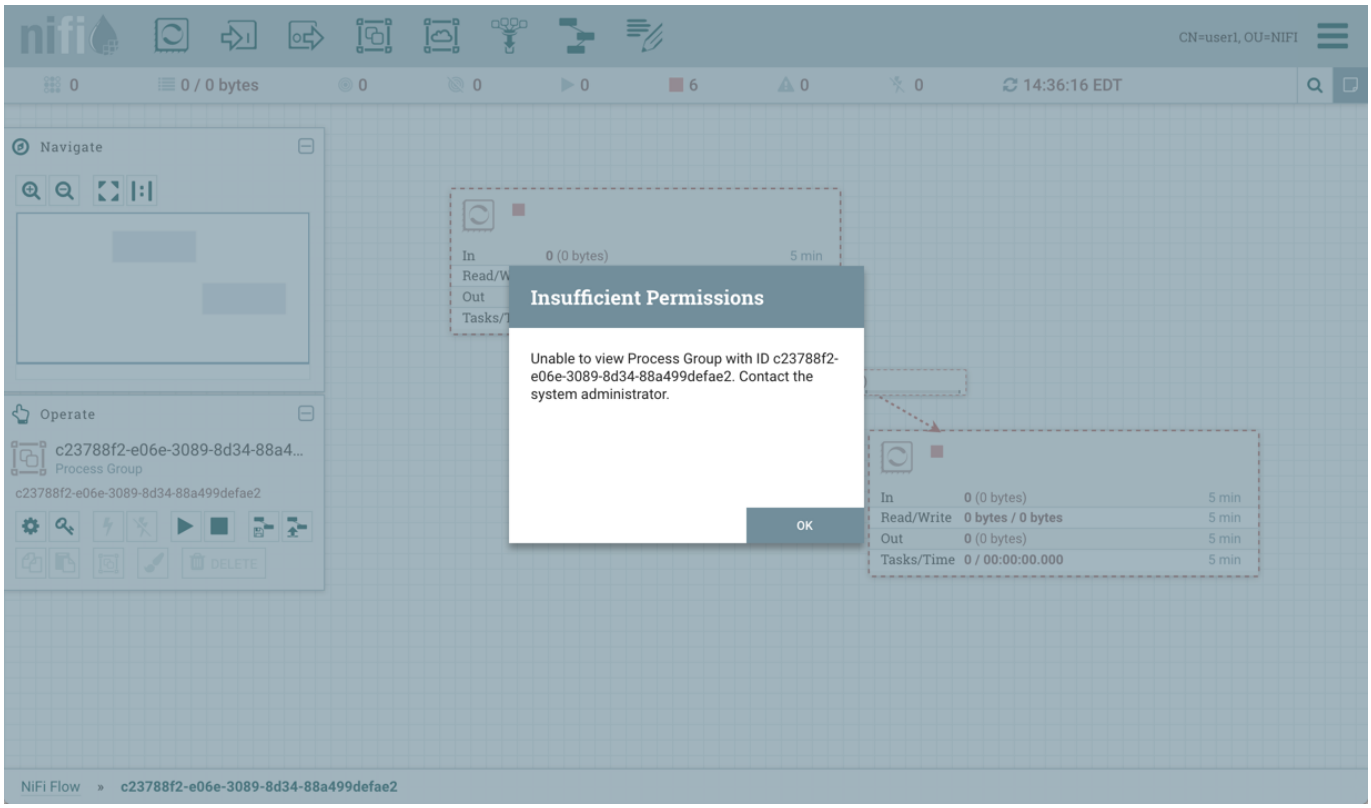


Рис.7.11.: Отсутствие доступа к экранной форме “Variables”

При выборе контроллера происходит переход к экранной форме сервиса окна конфигурации (Рис.7.13.).

7.1.5 Ссылки на компоненты

В случаях, когда компоненту, ссылающемуся на переменную, не предоставлены права на просмотр или изменение, в окне “Variables” отображается UUID данного компонента (Рис.7.14.).

В приведенном примере свойство *property1* ссылается на процессор, в котором у пользователя *user1* нет прав доступа на просмотр (Рис.7.15.).

7.2 Ссылка nifi.properties

Пользовательские свойства могут быть созданы и настроены с использованием ссылки *nifi.properties*. Для этого необходимо определить один или несколько наборов пар ключ/значение и передать их системному администратору. После добавления новых пользовательских свойств важно убедиться, что поле *nifi.variable.registry.properties* в файле *nifi.properties* обновлено.

Important: Для активации обновлений необходимо перезапустить NiFi

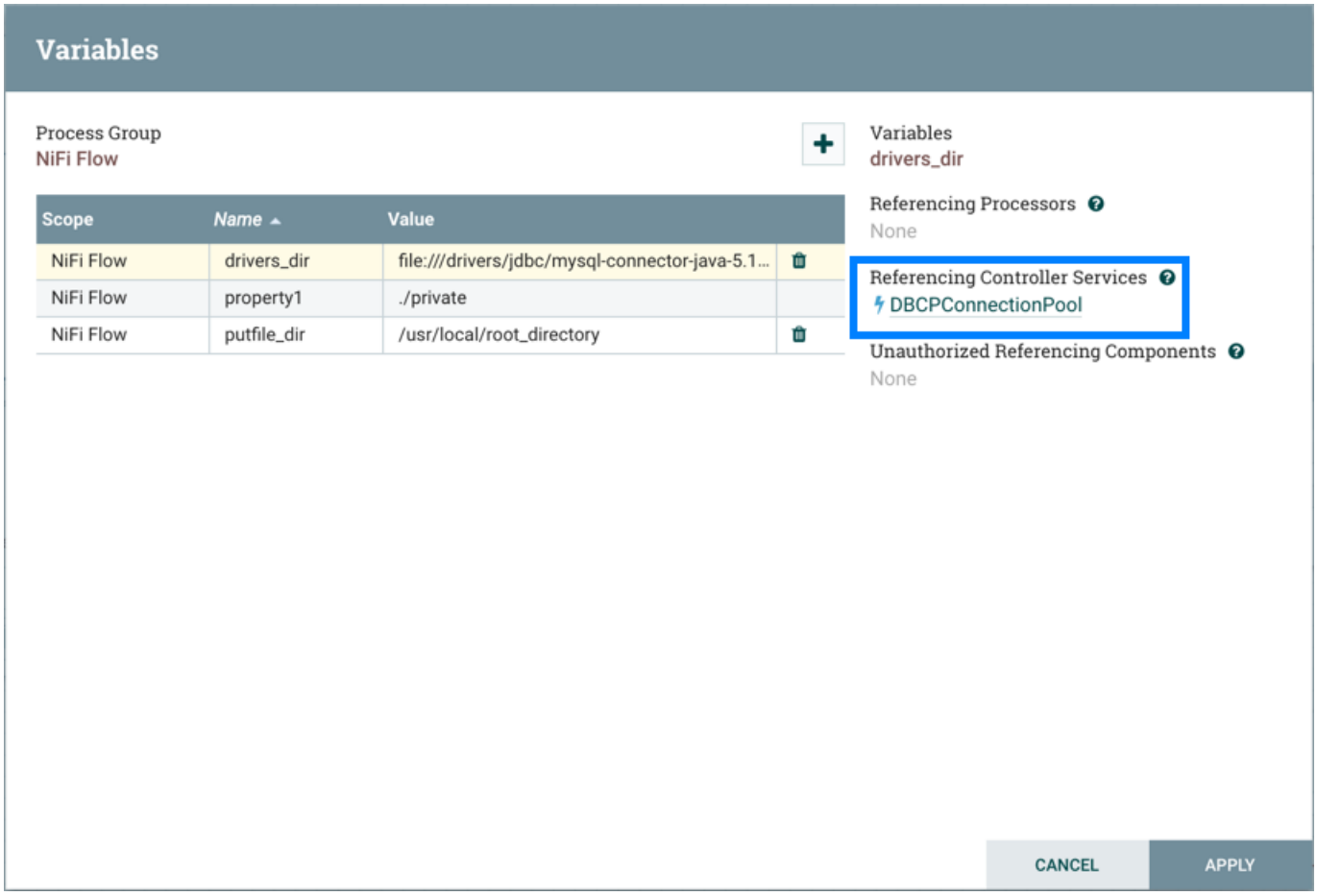


Рис.7.12.: Ссылка на Controller Services в “Variables”

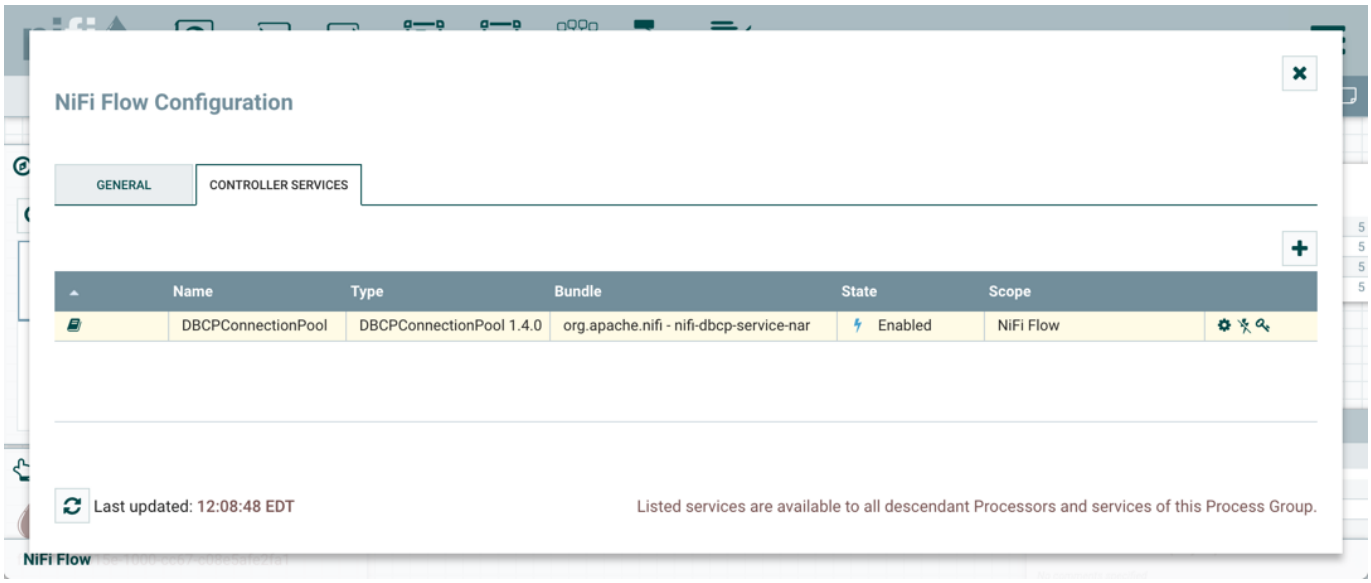


Рис.7.13.: Окно конфигурации Controller Services

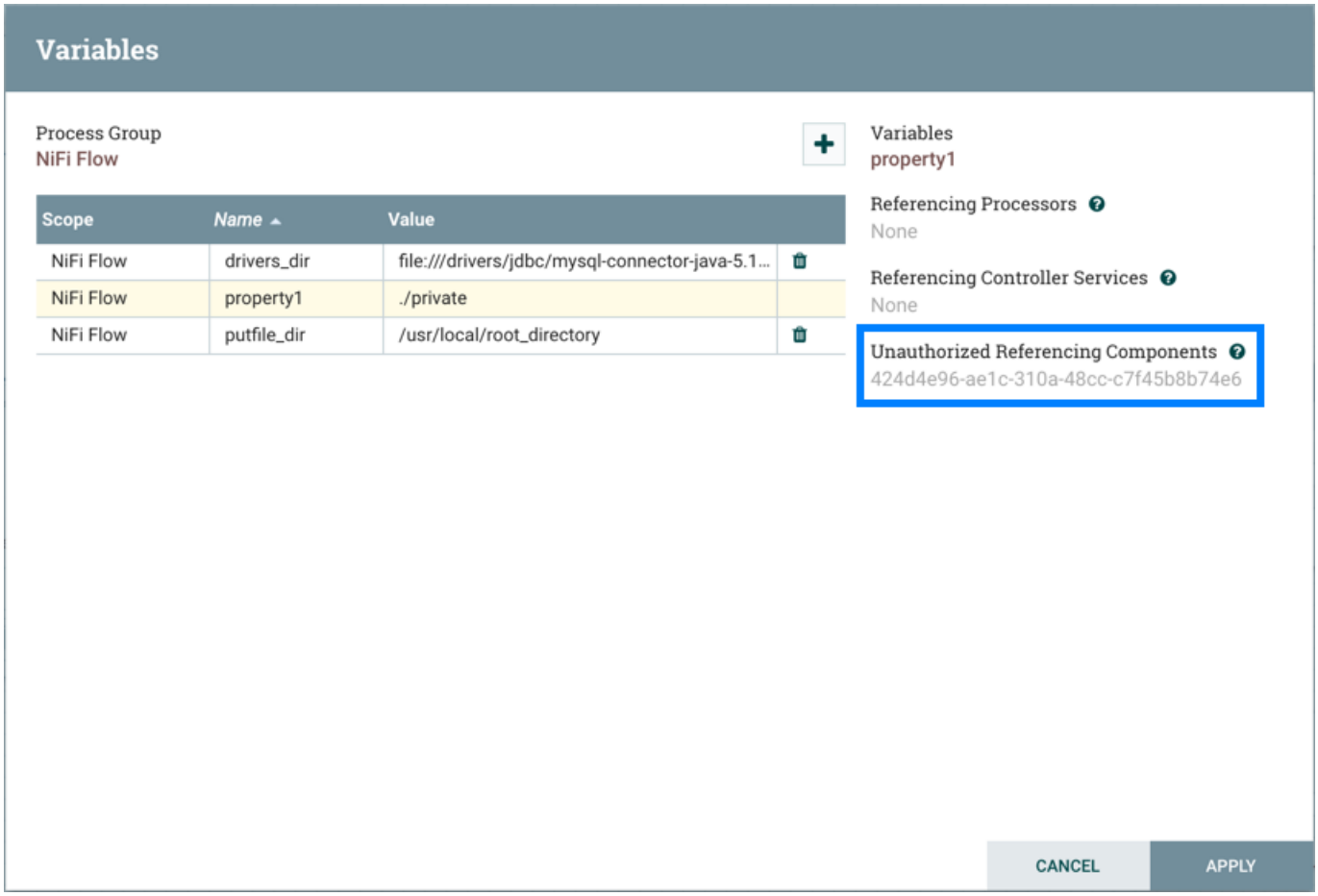


Рис.7.14.: Ссылка на UUID компонента в “Variables”

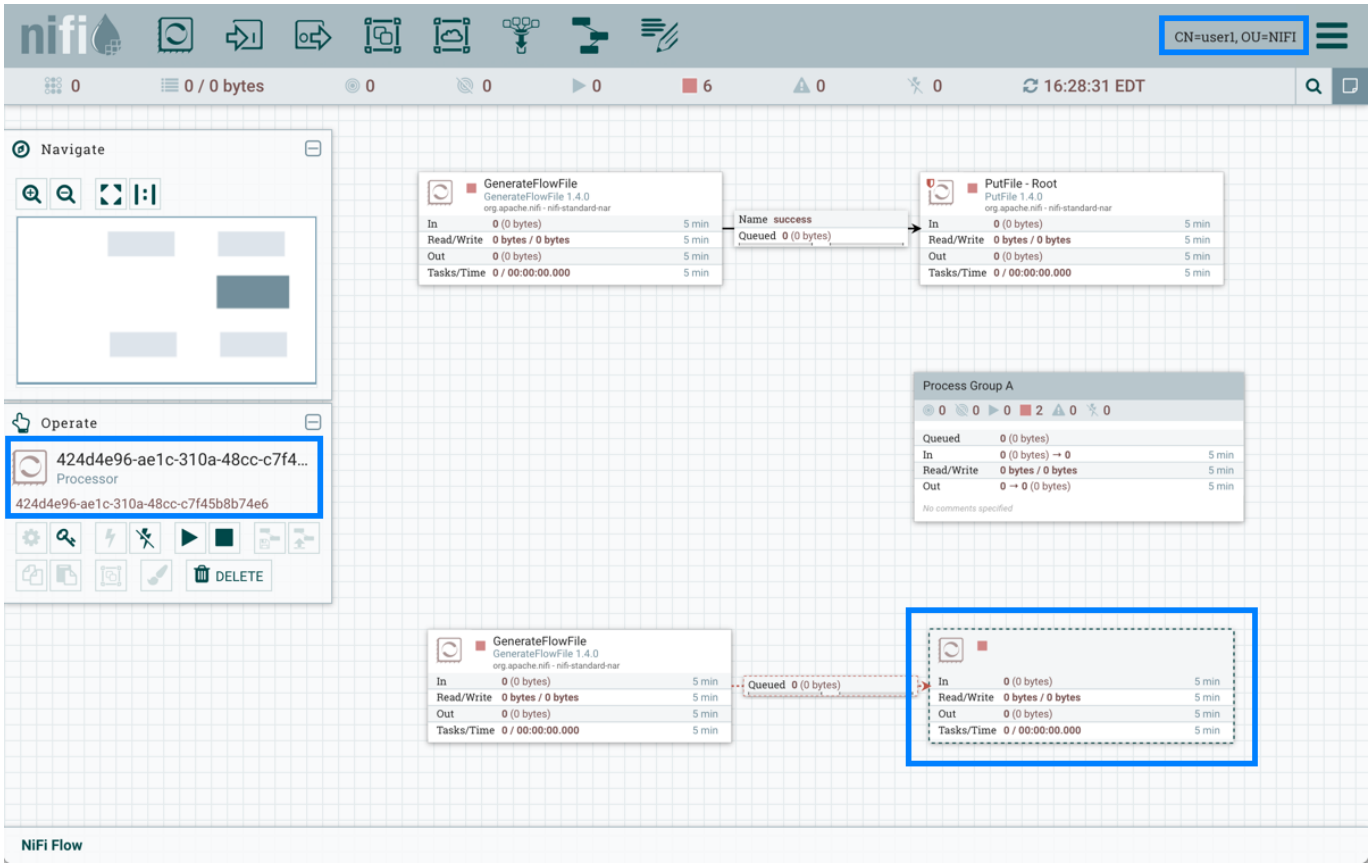


Рис.7.15.: Пример отсутствия прав доступа пользователя к компоненту

Глава 8

Управление потоком данных

При добавлении компонента в рабочую область **NiFi** он находится в статусе “Stopped”. Для перевода компонента в рабочее состояние его необходимо запустить. После запуска он может быть остановлен в любой момент времени. В статусе “Stopped” компонент можно настроить, запустить или отключить.

8.1 Запуск компонента

Для запуска компонента должны быть выполнены следующие условия:

- Конфигурация компонента действительна;
- Все определенные связи компонента подключены к другим компонентам или настроены на автоматическое завершение;
- Компонент остановлен;
- Компонент доступен;
- Компонент не имеет активных задач.

Для запуска отвечающего требованиям компонента необходимо выбрать его и нажать кнопку “Start” в палитре “Operate Palette” или щелкнув правой кнопкой мыши по компоненту и выбрать “Start” из открывшегося контекстного меню.

При запуске группы процессов запускаются все компоненты группы (включая дочерние), за исключением недействительных или отключенных.

После запуска индикатор состояния Процессора меняется на символ воспроизведения “Play”.

8.2 Остановка компонента

Компонент может быть остановлен в любой момент его выполнения. Для этого необходимо щелкнуть правой кнопкой мыши компонент и выбрать команду “Stop” из открывшегося контекстного меню или выбрать команду на панели “Operate Palette”. При остановке группы процессов останавливаются все компоненты группы (включая дочерние).

После остановки индикатор состояния компонента меняется на символ “Stop”.

Остановка компонента не прерывает его текущие задачи, а прекращает планирование выполнения новых задач. Количество активных задач отображается в правом верхнем углу Процессора.

8.3 Настройка компонента

Когда компонент доступен, он может быть запущен. При этом пользователи могут отключить компоненты, являющиеся частью потока данных, который все еще собирается. Как правило, если компонент не предназначен для запуска, он отключается, а не остается в состоянии “Stopped”. Это помогает различать компоненты, которые намеренно не запущены, и компоненты, которые могут быть временно остановлены (например, для изменения конфигурации компонента) и случайно не перезапущены.

При необходимости переподключения компонента это можно сделать, выбрав компонент и нажав кнопку “Enable” на панели “Operate Palette”. Это доступно только в том случае, когда выбранный компонент или компоненты отключены. Кроме того, компонент можно включить, установив флажок рядом с параметром “Enabled” на вкладке “Settings” диалогового окна “Processor configuration” или диалогового окна настройки порта.

Индикатор состояния отключенного компонента меняется на “Invalid” или “Stopped” в зависимости от того, является ли компонент действительным.

Компонент отключается путем выбора компонента и нажатия кнопки “Disable” на панели “Operate Palette” или путем снятия флажка рядом с параметром “Enabled” на вкладке “Settings” диалогового окна “Processor configuration” или диалогового окна настройки порта.

Включенными или отключенными компонентами могут быть только порты и процессоры.

8.4 Работа удаленных групп процессов

Удаленные группы процессов (Remote Process Group, RPG) предоставляют механизм отправки или извлечения данных из удаленного инстанса **NiFi**. Когда удаленная группа процессов добавляется в рабочую область, она добавляется с отключенной передачей со значком “Transmission Disabled” в верхнем левом углу. Передачу данных можно включить, щелкнув правой кнопкой мыши по RPG и выбрав пункт меню “Enable Transmission”. Это приводит к тому, что все порты, имеющие Соединение, начинают передачу данных, и индикатор состояния меняется на значок “Transmission Enabled”.

При возникновении проблем установления связи с удаленной группой процессов в верхнем левом углу появляется предупреждение “Warning”. Наведение курсора мыши на это предупреждение дает дополнительную информацию о проблеме.

8.4.1 Работа отдельного порта

В некоторых случаях DFM необходимо включить или отключить передачу данных только для определенного порта в группе удаленных процессов. Это можно сделать, щелкнув правой кнопкой мыши на RPG и выбрав пункт меню “Remote ports”. При этом открывается диалоговое окно настроек, в котором доступны конфигурации каждого порта (Рис.8.1.).

С левой стороны в блоке “Input Ports” перечислены все входные порты, на которые удаленный инстанс **NiFi** позволяет отправлять данные. Справа в блоке “Output Ports” – порты, из которых инстанс может извлекать данные. Если удаленный инстанс использует безопасную связь (его URL-адрес начинается с *https://*, а не *http://*), порты, которые он не сделал доступными, скрыты.

Если порт, который, как ожидается, должен быть показан, но при этом не отображается в диалоговом окне настроек портов, следует убедиться, что у инстанса есть надлежащие разрешения, и что поток в группе удаленных процессов является текущим. Проверить это можно, закрыв диалоговое окно “Port Configuration” и посмотрев в нижний правый угол группы удаленных процессов – отображается дата последнего обновления потока. Если поток является устаревшим, его можно обновить, щелкнув правой кнопкой мыши на RPG и выбрав “Refresh flow”.

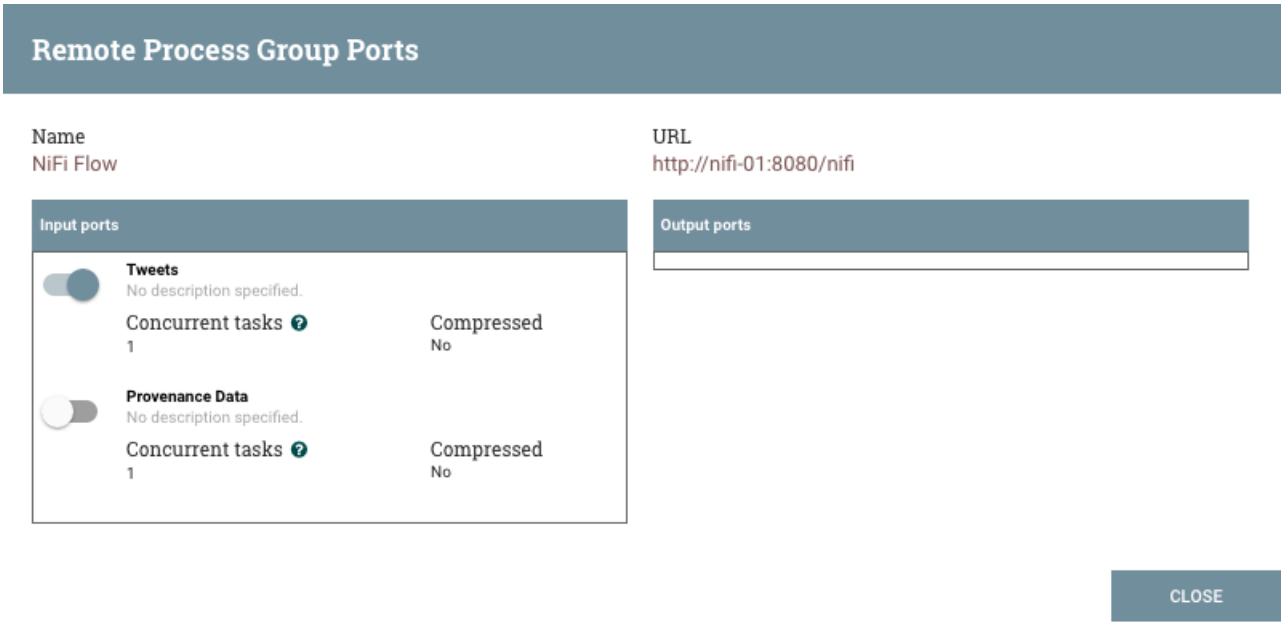


Рис.8.1.: Настройка отдельных портов в RPG

Каждый порт отображается с указанием его имени, сопровождающееся описанием, настроенным на данный момент количеством параллельных задач и необходимостью сжатия данных, отправляемых на порт. Слева от информации находится переключатель для включения или выключения порта. Порты, к которым не подключены соединения, выделены серым цветом (Рис.8.2.).

Переключатель предоставляет механизм для включения и отключения передачи данных для каждого порта в удаленной группе процессов независимо друг от друга. Неактивные в данный момент подключенные порты можно настроить, щелкнув значок карандаша “Edit” под переключателем состояния. Это позволяет DFM редактировать количество параллельных задач и определить, следует ли использовать сжатие при передаче данных посредством выбранного порта.

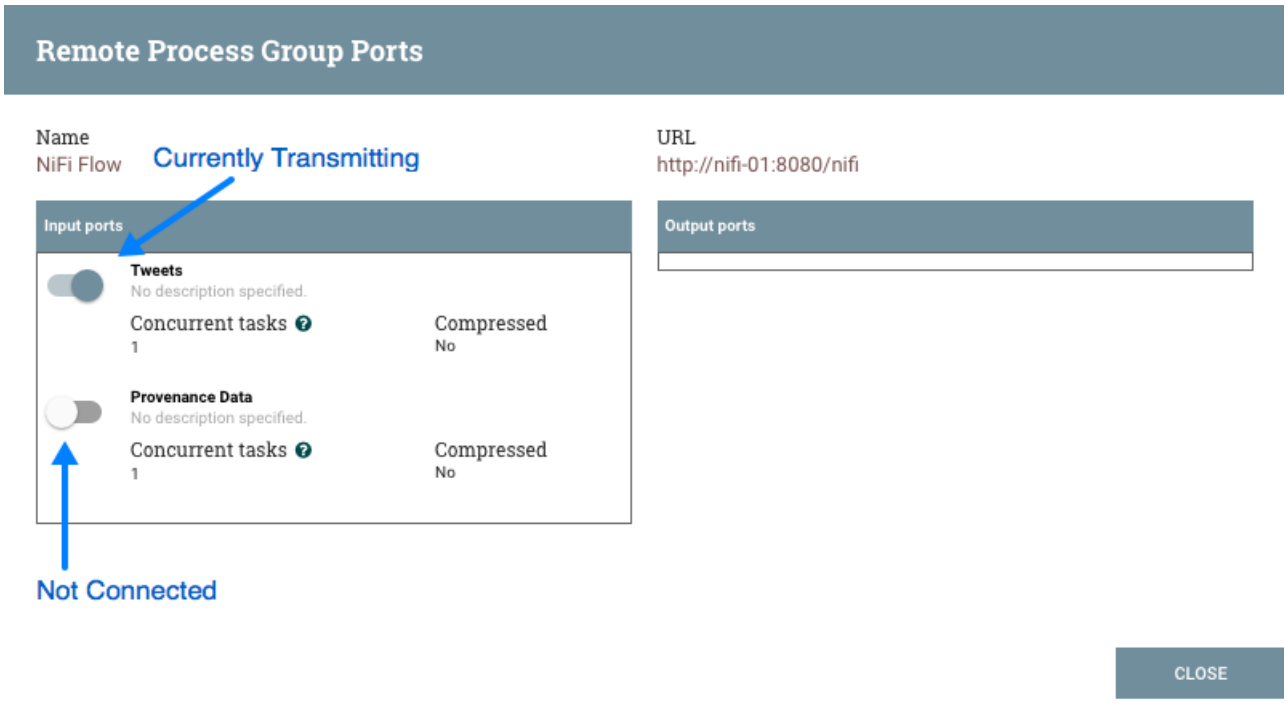


Рис.8.2.: Настройка отдельных портов в RPG