



Руководство администратора

Программный модуль

PhoenixDS Alarm Management

Аннотация

Программный модуль PhoenixDS Alarm Management (далее – ПМ PhoenixDS Alarm Management), расширяющий базовый функционал Программной платформы PhoenixDS (далее – ПП PhoenixDS), предназначен для автоматизации процессов оперативного реагирования на возникающие на производстве нештатные и аварийные ситуации.

ПМ PhoenixDS Operation Management используется в проектных решениях по построению систем управления производственными процессами (MES - Manufacturing Execution System).

Настоящее руководство администратора содержит описания принципа функционирования ПМ PhoenixDS Alarm Management, сведения о его настройке.



ДИНАМИЧЕСКИЕ СИСТЕМЫ

Россия, 101066, Москва, ул. Бауманская, д.53

тел.: +7 (495) 228-1100

www.dynasystems.ru

info@dynasystems.ru

Служба технической поддержки:

support@dynasystems.ru

Документ является собственностью ООО «Динамические Системы» и защищен законодательством о правах на результаты интеллектуальной деятельности. Никакая информация, содержащаяся в данном документе, не может быть воспроизведена, искажена, переработана, переведена на иностранный язык, записана или скопирована для любых коммерческих целей. Не допускается передача данного документа третьей стороне без письменного согласия ООО «Динамические Системы». Графические изображения и названия продуктов, упоминаемые в данном документе, могут быть зарегистрированными товарными знаками, охраняемыми законодательством о правах на результаты интеллектуальной деятельности. ООО «Динамические Системы» оставляет за собой право вносить изменения в содержание данного документа без предварительного уведомления.

Содержание

1. Введение	5
1.1. Назначение ПМ PhoenixDS Alarm Management.....	5
2. Понятия и концепции	6
2.1. Сигналы и оповещения.....	6
2.1.1. Основные определения	6
2.1.2. Глобальные и локальные значения параметров	8
2.1.3. Логическая схема вычисления статусов сигналов и отправки оповещений	9
2.1.4. Параметры сигналов и алгоритмы определения статуса	11
2.1.5. Класс оповещений Simple	17
2.1.6. Методы и типы оповещений	19
2.1.7. Универсальный шаблон оповещений	20
3. Настройка ПМ PhoenixDS Alarm Management.....	22
3.1. Настройка СУБД ClickHouse для хранения данных о сигналах	22
3.1.1. Установка и настройка Apache ZooKeeper	22
3.1.2. Настройка кластера ClickHouse	23
3.1.3. Настройка таблиц в БД ClickHouse	24
3.1.4. Переключение ПМ PhoenixDS Alarm Management	25
3.1.5. Миграция данных из БД MySQL в БД ClickHouse	26
4. Администрирование ПМ PhoenixDS Alarm Management.....	27
4.1. Редактор сигналов и оповещений (обзор web-приложения).....	27
4.1.1. Панель инструментов и контекстное меню	29
4.1.2. Вкладка Сигналы	30
4.1.3. Вкладка Оповещения	31
4.2. Настройка сигналов	32
4.2.1. Настройка наборов сигналов.....	34
4.2.2. Назначение наборов сигналов целевым объектам	43
4.3. Настройка оповещений.....	49
4.3.1. Настройка наборов оповещений	51
4.3.2. Назначение наборов оповещений целевым объектам	58
4.3.3. Настройка универсальных шаблонов оповещений	63
4.4. Редактор информационных сообщений.....	67
4.4.1. Основное окно.....	68
4.4.2. Создание шаблона информационных сообщений	69
4.4.3. Основные параметры шаблона информационных сообщений.....	70
4.4.4. Добавление дополнительных параметров в шаблон.....	72
4.4.5. Синтаксис HTML-шаблона	73
4.4.6. Тестирование шаблона информационных сообщений	76
4.4.7. Отправка информационного сообщения вручную.....	77
5. Справочники.....	78

5.1.	Системные процессы ПМ PhoenixDS Alarm Management.....	78
5.1.1.	Формирование сигналов (alerts).....	78
5.1.2.	Формирование оповещений (notifier)	80

1. ВВЕДЕНИЕ

1.1. Назначение ПМ PhoenixDS Alarm Management

ПМ **PhoenixDS Alarm Management** предназначен для автоматизации процессов оперативного реагирования на возникающие на производстве нештатные и аварийные ситуации. Своевременное обнаружение неординарных событий и четкое следование регламентам и инструкциям является определяющим фактором оперативной локализации и предотвращения дальнейшего развития нештатных и аварийных ситуаций, опасных для персонала, окружающей среды и технологического оборудования.

ПМ **PhoenixDS Alarm Management** поддерживает выполнение следующих задач:

- ❖ регистрация нештатных ситуаций;
- ❖ проведение классификации, оценки и определения категории важности событий;
- ❖ оповещение ответственных специалистов предприятия в соответствии с утвержденным регламентом;
- ❖ контроль развития нештатных и аварийных ситуаций;
- ❖ хранение архива событий для расследования инцидентов в различных форматах.

2. ПОНЯТИЯ И КОНЦЕПЦИИ

2.1. Сигналы и оповещения

ПП PhoenixDS предоставляет следующие функциональные возможности:

- ❖ формирование оповещений о проблемах с использованием различных методов. Например, одно и то же оповещение может быть отправлено по электронной почте, SMS, а также записано в лог для последующего анализа;
- ❖ отправка оповещений о проблемах на каждом целевом объекте для разных групп пользователей. Например, группе администраторов необходимо получать оповещения от всех сетевых интерфейсов, но отдельному оператору достаточно получать оповещения только о проблемах непосредственно на его линиях;
- ❖ контроль времени отправки сообщения. Например, можно настроить повторную отправку оповещения через 4 часа после первой отправки, если проблема так и не была решена.

За функционирование механизма оповещений о событиях отвечают системные процессы [alerts](#) и [notifs](#). Прежде, чем детально описывать их алгоритмы, необходимо ввести ряд понятий.

2.1.1. Основные определения

Схема показывает фрагмент дерева конфигурации ПП PhoenixDS и иллюстрирует структурные связи между сущностями, которыми оперирует системный процесс `alerts`. Далее детально рассматривается назначение каждой сущности.

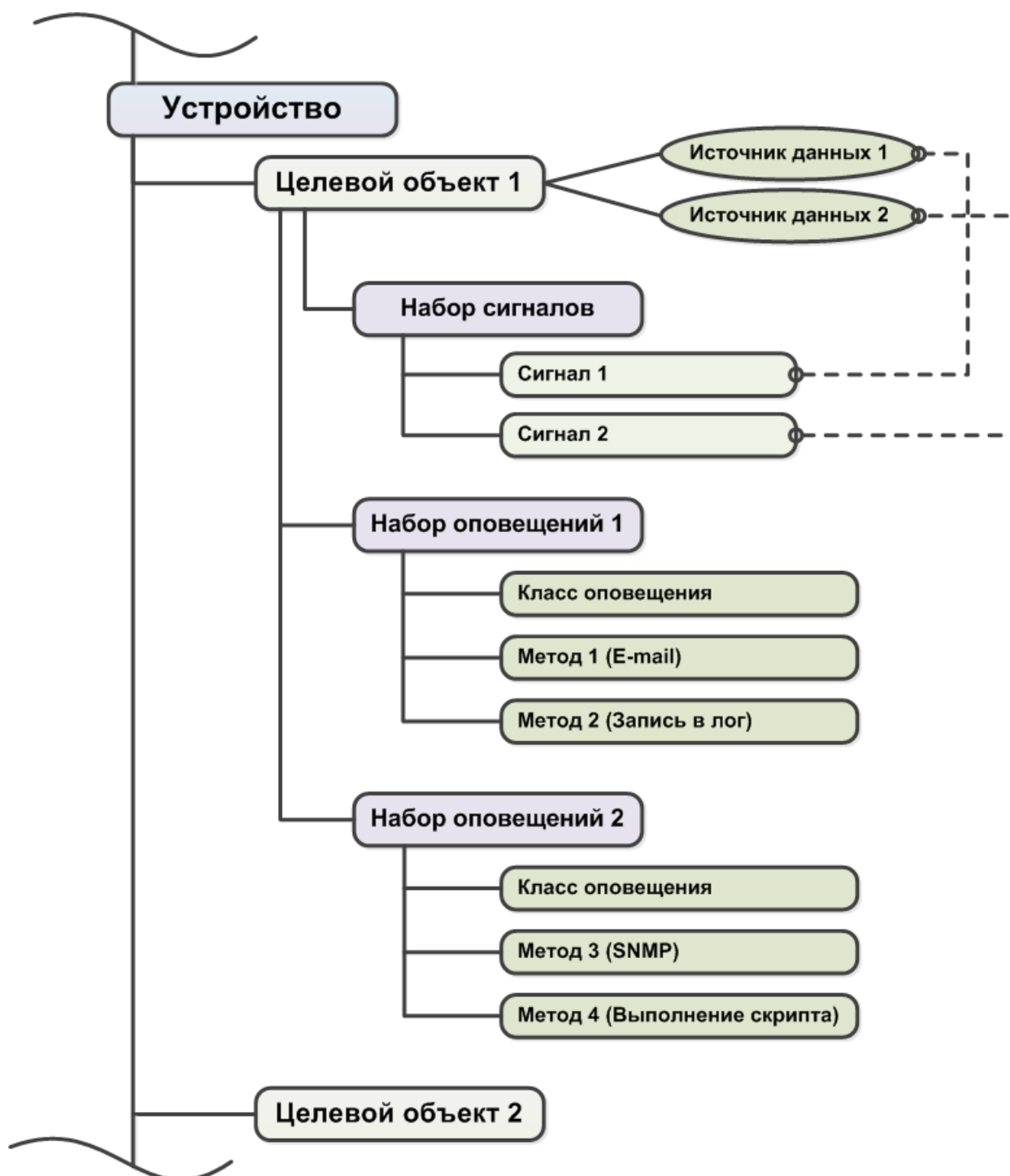


Рисунок 1. Схема привязки сущностей, используемых системным процессом alerts

Сигнал и набор сигналов

Сигнал – правило обработки значений параметров целевых объектов и определения статуса сигнала. Параметры целевых объектов, которые анализируются правилом, также называются *источниками данных*. Ключевым атрибутом сигнала является тип сигнала.

Тип сигнала – алгоритм вычисления статуса сигнала, ограничивающий множество возможных статусов сигнала.

Значение сигнала – числовое значение, полученное в результате обработки значений параметров целевых объектов за определенный интервал времени (например: среднее значение, максимальное или минимальное значение). Значение сигнала используется для вычисления статуса сигнала.

Статус сигнала – характеристика сигнала, отражающая наличие проблемы на целевом объекте и уровень ее критичности. Статус сигнала идентифицируется цветом и числовой константой.

См. также справочник Статусы сигналов.

Событие – изменение статуса сигнала.

Длительность события – интервал времени между двумя последовательными изменениями статуса сигнала.

При настройке сигнала задаются:

- ❖ период времени и функция для определения **значения сигнала**;
- ❖ параметры алгоритма для вычисления **статуса сигнала** (пороговые значения).

Набор сигналов – совокупность сигналов, которые применимы к источникам данных, принадлежащим целевым объектам определенного типа. Количество сигналов в наборе ограничивается количеством источников данных, поддерживаемых ПП PhoenixDS для выбранного типа целевого объекта.

Таким образом, для создания набора сигналов необходимо ввести его общие параметры (наименование и тип целевого объекта) и составить множество сигналов.

Локальный администратор привязывает наборы сигналов к целевым объектам. К конкретному целевому объекту можно привязать только один набор сигналов. Однако один и тот же набор сигналов можно привязать к нескольким целевым объектам.

Класс оповещений, метод оповещений, набор оповещений

Оповещение – это уведомление (пользователей) о произошедшем событии (например, сбое, изменении статуса параметра объекта контроля) и /или фиксирование таких фактов в системе (например, логирование).

Класс оповещений – совокупность параметров, определяющих условия, при возникновении которых формируются оповещения. В частности, в классе оповещения указывается, по каким сигналам и при каких статусах сигналов формируется оповещение, а также алгоритм формирования повторных оповещений.

Метод оповещений – совокупность параметров, определяющих способ формирования оповещений. Метод оповещений содержит ключевой параметр **тип оповещения** и параметры, конкретизирующие выполняемое действие, например: тип оповещения «отправка сообщения по электронной почте», адреса получателей сообщений, сетевой адрес почтового сервера.

Тип оповещения – вид уведомления, например, отправка сообщения по электронной почте или SMS, создание записи в лог-файле или базе данных, передача сообщения о событии в стороннюю систему, выполнение иной программной процедуры (скрипта). Тип оповещения определяет способ формирования и доставки оповещений.

Набор оповещений – совокупность класса и метода (или нескольких методов) оповещений. Набор оповещений определяет, при каких условиях должно выполняться то или иное действие.

Набор оповещений обязательно должен содержать только один класс оповещений. В состав одного набора могут быть включены несколько методов оповещений, причем в любой комбинации способов создания и отправки. Например, набор может содержать 2 метода для отправки оповещения по электронной почте, 2 метода для отправки sms и один метод для записи оповещения в лог.

Локальный администратор привязывает наборы оповещений к целевым объектам, также как и наборы сигналов. Однако, в отличие от наборов сигналов, к конкретному целевому объекту можно привязать несколько наборов оповещений. Один и тот же набор оповещений можно привязать к нескольким целевым объектам.

2.1.2. Глобальные и локальные значения параметров

При настройке набора сигналов и набора оповещений для каждого его структурного элемента (т.е. для сигнала, класса оповещений, метода оповещений) администратор задает **глобальные значения** параметров. На схеме уровень глобальных значений показан **синим** цветом.

При привязке набора к конкретному целевому объекту для каждого структурного элемента пользователь может задать **локальные значения** параметров. На схеме ниже уровень локальных значений показан **красным** цветом. Локальное значение действует только в рамках конкретного целевого объекта и имеет повышенный приоритет по сравнению с глобальным значением:

- ❖ если для параметра задано локальное значение, то системный процесс `alerts` использует именно его, игнорируя глобальное значение;
- ❖ если для параметра локальное значение не определено, то системный процесс `alerts` использует глобальное значение.

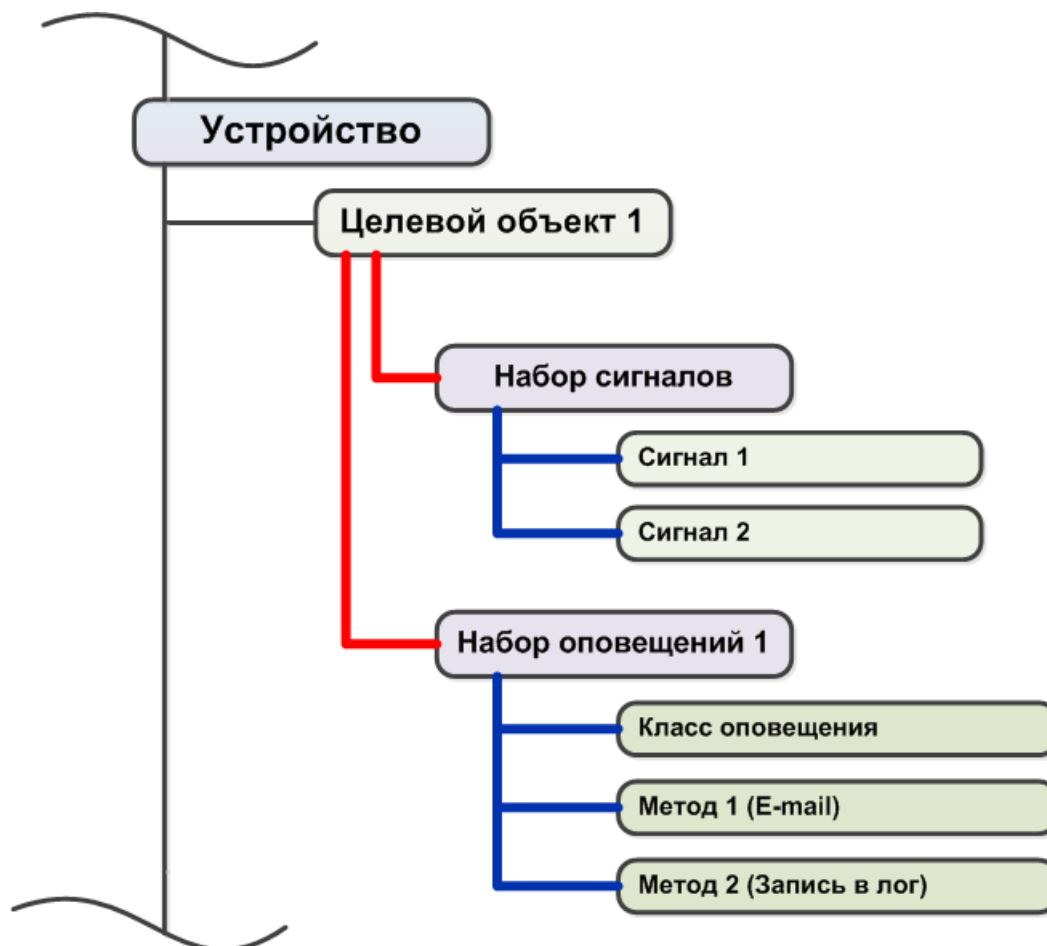


Рисунок 2. Схема привязки наборов сигналов и оповещений

2.1.3. Логическая схема вычисления статусов сигналов и отправки оповещений

В ПМ **PhoenixDS Alarm Management** за обработку данных, собранных с объектов контроля, и вычисление статусов сигналов отвечает системный процесс [alerts](#), за формирование оповещений – системный процесс [notifs](#). Рассмотрим схему их работы на примере.

Предположим, что к целевому объекту 1 с двумя источниками данных привязаны: набор сигналов, состоящий из двух сигналов для разных типов источников данных, и два набора оповещений.

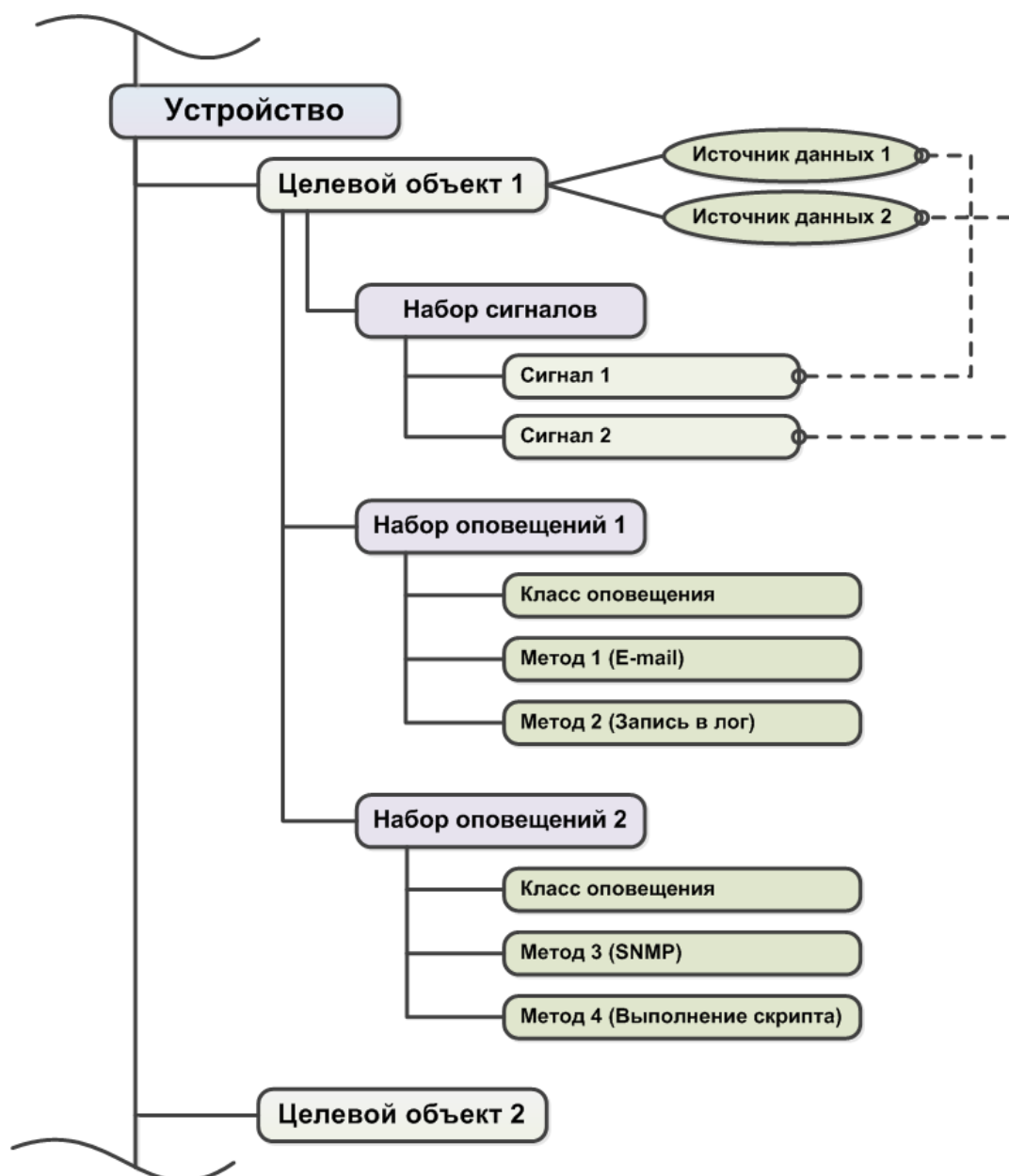


Рисунок 3. Схема привязки сущностей, используемых сервисом alerts

На схеме ниже зеленым цветом выделены элементы и стрелки, отражающие обработку значений от источника данных 1, синим цветом – от источника данных 2.

Обработка значений от источников данных и формирование оповещений осуществляется следующим образом:

1. системный процесс `alerts` обрабатывает каждый сигнал из набора сигналов применительно к источнику данных соответствующего типа:
 - ❖ сначала определяется значение сигнала – для этого к множеству значений, полученных от источника данных за определенный период, применяется функция (например, функция выбирает минимальное значение или рассчитывает среднее значение);
 - ❖ затем вычисляется статус сигнала – для этого используется определенный алгоритм (например, значение сигнала сравнивается с пороговыми значениями);
2. системный процесс `notifs` обрабатывает наборы оповещений применительно к каждому сигналу в рамках целевого объекта:
 - ❖ для каждого сигнала определяется необходимость формирования оповещения на основании условий, описанных в классе оповещения;
 - ❖ если такая необходимость выявлена, то оповещения формируются в соответствии с каждым из методов, заведенных в наборе оповещений.

На схеме оповещения по сигналу 1 сформированы в соответствии с наборами оповещений 1 и 2 (зеленые стрелки), оповещения по сигналу 2 – только в соответствии с набором оповещений 2 (синие стрелки).

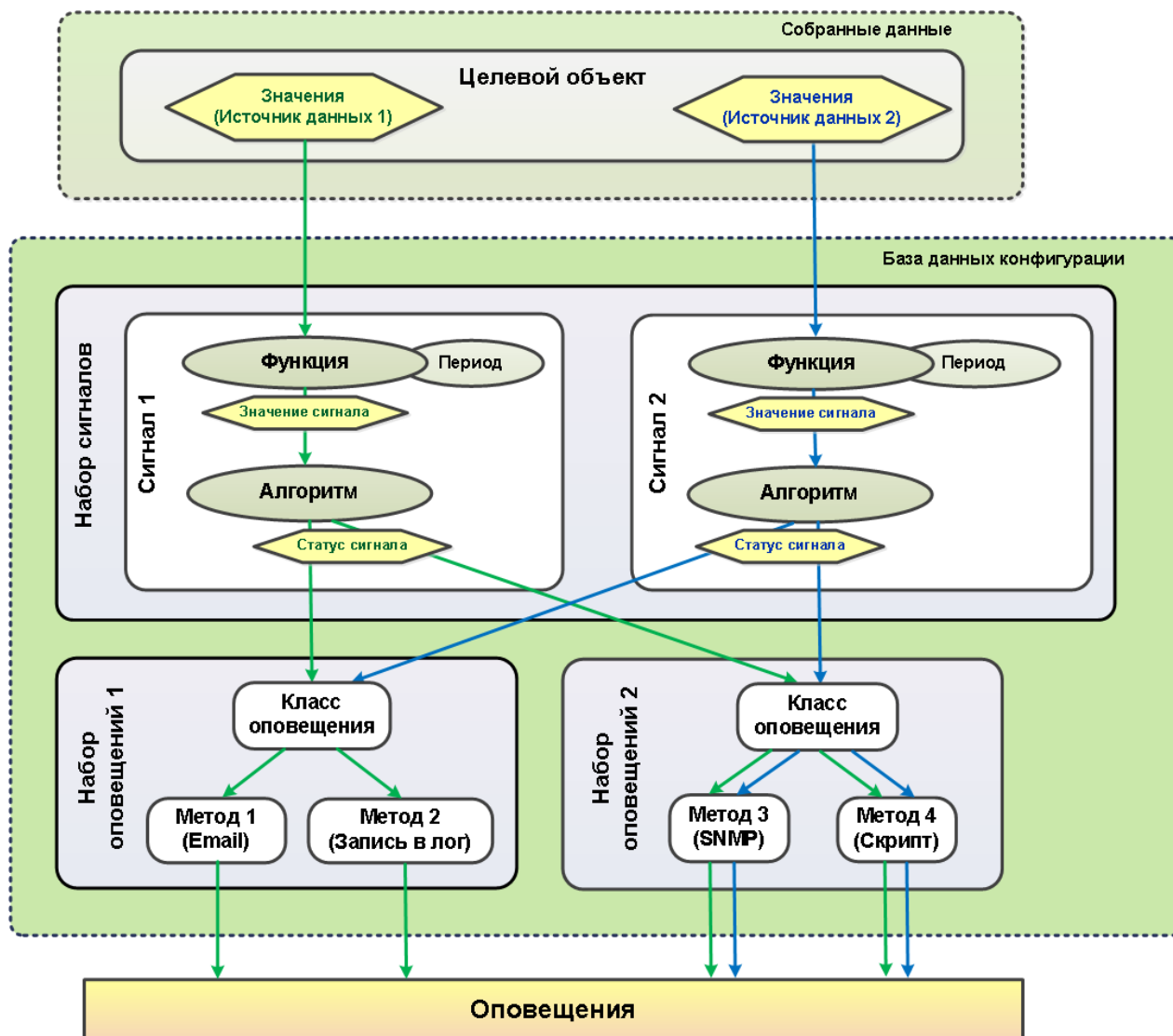


Рисунок 4. Схема функционирования системных процессов alerts и notifys

Значение сигнала и его статус пользователь может просмотреть в Web-интерфейсе. Оба параметра используются также модулем **PhoenixDS Dashboards** при построении информационных панелей.

2.1.4. Параметры сигналов и алгоритмы определения статуса

При добавлении сигнала в набор сигналов администратор задаёт следующие параметры:

- ❖ наименование сигнала;
- ❖ тип сигнала – определяет алгоритм вычисления статуса сигнала и множество возможных статусов сигнала; реализованы следующие типы сигналов:
 - [Gauge](#);
 - [InvertGauge](#);
 - [SmartGauge](#);
 - [Availability](#);
- ❖ наименование источника данных;
- ❖ [функция и период для определения значения сигнала](#);
- ❖ параметры алгоритма вычисления статуса сигнала;
- ❖ [признак лидирующего сигнала](#) (для сигналов типа [Gauge](#) и [Availability](#)).

В приведенном списке первые три параметра являются ключевыми, т.к. именно они определяют принципы работы сигнала. Поэтому эти параметры имеют только [глобальные](#) значения. Для остальных параметров сигнала могут быть заданы [локальные](#) значения.

2.1.4.1. Функция и период для определения значения сигнала

Значение сигнала определяется с помощью функции, которая выбирает из базы данных производительности (ClickHouse или RRD) одно из значений, полученных от источника данных за заданный период времени **Диапазон** (*range*).

Доступны следующие функции:

- ❖ **среднее** (*avg*) – выбирает из базы данных значения за интервал **Диапазон** (*range*) и вычисляет среднее значение;
- ❖ **минимальное** (*min*) – выбирает из базы данных минимальное значение за интервал **Диапазон** (*range*);
- ❖ **максимальное** (*max*) – выбирает из базы данных максимальное значение за интервал **Диапазон** (*range*);
- ❖ **последнее** (*last*) – выбирает из базы данных последнее известное значение за интервал **Диапазон** (*range*). Это значение может быть только числом;
- ❖ **текущее** (*cur*) – выбирает из базы данных последнее значение за интервал **Диапазон** (*range*). Это значение не обязательно будет числом;

Например, если в течение интервала **Диапазон** (*range*) целевой объект стал недоступным, то функция *last* возвращает значение, полученное в момент, когда целевой объект был еще доступен, а функция *cur* возвращает значение **NAN**;

- ❖ **последнее известное** (*ldef* – от last defined) – выбирает из базы данных последнее известное значение за интервал **Диапазон** (*range*). Если за указанный интервал значения отсутствуют или они не определены, то функция возвращает значение **NAN**;
- ❖ **не определено** – эквивалент незаданному значению; допустимо только как локальное значение параметра **Функция** для указания на использование глобального значения.

Период времени **Диапазон** (*range*) задаётся в секундах. Чем он больше, тем больше времени требуется для вычисления значения сигнала. По умолчанию – 600 секунд.

2.1.4.2. Алгоритм определения статуса сигнала типа Gauge

Сигнал типа **Gauge** характеризует уровень использования/загрузки интерфейса или другого ресурса (общий принцип: чем больше значение сигнала, тем "хуже" его статус).

Сигналу присваивается статус в соответствии с диапазоном, в который попадает значение сигнала. Диапазоны определяются пороговыми значениями, которые задает администратор в качестве глобальных или локальных значений параметров сигнала:

- ❖ **Нижний порог** (*low*) – по умолчанию равен 0. Как правило, низкий порог не указывают в явном виде, но при необходимости пользователь может его задать в абсолютных единицах;
- ❖ **Нормальный порог** (*normal*) – задается пользователем в процентах от максимального порога или в абсолютных единицах;
- ❖ **Высокий порог** (*high*) – задается пользователем в процентах от максимального порога или в абсолютных единицах;
- ❖ **Максимальный порог** (*max*) – по умолчанию равен значению параметра *Bandwidth* (пропускная способность), определенному для целевого объекта при регистрации устройства в дереве конфигурации. Если параметр *Bandwidth* не определен, то по умолчанию максимальный порог принимается равным **100**. Как правило, максимальный порог не указывают в явном виде, но при необходимости пользователь может задать его в абсолютных единицах.

В таблице приведен перечень статусов сигналов типа `Gauge`, и для каждого статуса указано условие его присвоения – диапазон значений сигнала в привязке к пороговым величинам.

Цветовой код статуса (по умолчанию)	Описание статуса	Условие присвоения статуса (x – значение сигнала)
Серый (grey)	Состояние неизвестно. Устройство не отвечает на запросы.	x – не определено (NaN)
Синий (blue)	Значение сигнала меньше нормы.	$x < low$
Зелёный (green)	Нормальное состояние.	$low \leq x < normal$
Жёлтый (yellow)	Предупреждение: значение сигнала превысило норму.	$normal \leq x < high$
Красный (red)	Проблема: значение сильно превысило норму.	$high \leq x < max$
Бордовый (crimson)	Критическое состояние: значение сигнала превышает пропускные возможности.	$max \leq x$

Алгоритм определения статуса сигнала проиллюстрирован на рисунке.



Рисунок 5. Алгоритм определения статуса сигнала типа `Gauge`

См. также:

Параметры сигнала типа `Gauge`

2.1.4.3. Алгоритм определения статуса сигнала типа `InvertGauge`

Сигнал типа `InvertGauge` характеризует уровень неиспользованного ресурса, например тонера для принтера, объема свободного места на диске (общий принцип: чем меньше значение сигнала, тем "хуже" его статус).

Сигналу присваивается статус в соответствии с диапазоном, в который попадает значение сигнала. Диапазоны определяются пороговыми значениями, которые задает администратор в качестве глобальных или локальных значений параметров сигнала:

- ❖ **Высокий порог** (`high`) – по умолчанию принимается равным **100**;
- ❖ **Нормальный порог** (`normal`) – задается пользователем в процентах от верхнего порога или в абсолютных единицах;
- ❖ **Низкий порог** (`low`) – по умолчанию равен 0. Как правило, низкий порог не указывают в явном виде, но при необходимости пользователь может его задать в абсолютных единицах;

- ❖ **Минимальный порог** (\min) – по умолчанию принимается равным **0**. Как правило, минимальный порог не указывают в явном виде, но при необходимости пользователь может задать его в процентах от верхнего порога или в абсолютных единицах.

В таблице приведен перечень статусов сигналов типа *InvertGauge*, и для каждого статуса указано условие его присвоения – диапазон значений сигнала в привязке к пороговым величинам.

Цветовой код статуса (по умолчанию)	Описание статуса	Условие присвоения статуса (x – значение сигнала)
Серый (grey)	Состояние неизвестно. Устройство не отвечает на запросы.	x – не определено (NaN)
Синий (blue)	Значение превышает норму.	$x \geq \text{high}$
Зелёный (green)	Нормальное состояние.	$\text{normal} \leq x < \text{high}$
Жёлтый (yellow)	Предупреждение: значение приближается к критическому.	$\text{low} \leq x < \text{normal}$
Красный (red)	Проблема: значение опустилось ниже критического.	$\min \leq x < \text{low}$
Бордовый (crimson)	Критическая проблема: значение ниже минимального.	$x < \min$

Алгоритм определения статуса сигнала проиллюстрирован на рисунке.



Рисунок 6. Алгоритм определения статуса сигнала типа *InvertGauge*

См. также:

Параметры сигнала типа *InvertGauge*

2.1.4.4. Алгоритм определения статуса сигнала типа *SmartGauge*

Сигнал типа **SmartGauge** характеризует состояние сигнала в соответствии с заданными диапазонами значений. Администратор может настроить произвольное количество диапазонов (в этом принципиальное отличие от сигнала типа **Gauge** и **InvertGauge**).

Сигналу присваивается статус в соответствии с диапазоном, в который попадает значение сигнала. Бордовый статус не предусмотрен. Для каждого статуса может быть задано несколько диапазонов. Если значение сигнала не попадает ни в один из заданных диапазонов, то сигналу присваивается синий статус.

ВНИМАНИЕ! Логика разграничения диапазонов лежит в зоне ответственности администратора. Он настраивает границы диапазонов в соответствии с техническими характеристиками объектов контроля и особенностями технологических процессов предприятия.

Таким образом, для сигнала типа **SmartGauge** задаются следующие параметры:

- ❖ **Нижний порог** – диапазон значений для синего статуса сигнала;
- ❖ **Нормальный диапазон** – диапазон значений для зеленого статуса сигнала;
- ❖ **Незначительный порог** – диапазон значений для желтого статуса сигнала;
- ❖ **Высокий порог** – диапазон значений для красного статуса сигнала;
- ❖ **Максимальный порог** – значение, от которого берутся проценты при вычислении границ диапазонов. По умолчанию равен значению параметра `Bandwidth` (пропускная способность), определенному для целевого объекта при регистрации устройства в дереве конфигурации. Если параметр `Bandwidth` не определен, то по умолчанию максимальный порог принимается равным **100**. Как правило, максимальный порог не указывают в явном виде, но при необходимости пользователь может задать его в абсолютных единицах.

При настройке диапазонов следует руководствоваться правилами:

- ❖ диапазон вводится в формате: **[a:b]**, причем нижняя граница принадлежит диапазону ($a \leq x < b$);
- ❖ для каждого статуса можно задать несколько диапазонов, перечислив их через запятую;
- ❖ диапазон можно задать в абсолютных единицах или в процентах от **Максимального порога**. В последнем случае символ **%** указывается однократно после списка диапазонов, например: **[10:20],[70:80]%**;
- ❖ для указания бесконечности следует использовать обозначения **-inf** и **inf**. Один из диапазонов обязательно должен быть не ограничен справа, например: **[90:inf]%**.

Значение сигнала



Рисунок 7. Настройки по умолчанию для алгоритма определения статуса сигнала типа **SmartGauge**

См. также:

Параметры сигнала типа **SmartGauge**

2.1.4.5. Алгоритм определения статуса сигнала типа **Availability**

Сигнал типа **Availability** показывает доступность целевого объекта, а именно: интерфейса, и используется для выявления недоступных интерфейсов.

Алгоритм определения статуса сигнала основан на булевой логике и использует эталонную величину (*ideal*):

- ❖ если значение сигнала совпадает с эталонной величиной то, сигнал получает **зеленый** статус;
- ❖ при любом другом значении сигнала – **красный** статус.

В таблице приведен перечень статусов сигналов типа *Availability*, и для каждого статуса указано условие его присвоения.

Цветовой код статуса (по умолчанию)	Описание статуса	Условие присвоения статуса (x – значение сигнала)
Зелёный (green)	Проблем нет. Интерфейс работает.	$x = ideal$
Красный (red)	Проблема требует внимания. Интерфейс не работает.	$x \neq ideal$
Серый (grey)	Состояние неизвестно. Устройство не отвечает на запросы.	x – не определено

Таким образом, для сигнала типа *Availability* предусмотрены следующие настраиваемые параметры:

- ❖ **Эталон** (*ideal*) – эталонная величина;
- ❖ **Описание для "хороший"** (*green legend*) – текст, отображаемый в оповещении при **зеленом** статусе сигнала;
- ❖ **Описание для "плохой"** (*red legend*) – текст, отображаемый в оповещении при **красном** статусе сигнала.

См. также:

Параметры сигнала типа *Availability*

2.1.4.6. Признак Лидер

Лидер (Leader) – признак, свидетельствующий о повышенном приоритете сигнала в наборе сигналов. Признак является параметром сигналов типов *Gauge*, *InvertGauge*, *SmartGauge* и *Availability*.

По умолчанию ни один сигнал не является лидером (лидирующим). Признак **Лидер** может быть установлен для одного или нескольких сигналов в наборе.

Если в наборе, привязанном к целевому объекту, нет лидирующих сигналов, то всегда обрабатываются все сигналы, т.к. они считаются равнозначными. В этом случае, может возникнуть ситуация, при которой сразу все сигналы получают «плохой» статус, например, если целевой объект стал недоступен. При соответствующих настройках набора оповещений по каждому сигналу может быть сформировано оповещение (или несколько оповещений); а если спустя некоторое время проблемы не были устранены, то могут отправляться еще и повторные оповещения.

Чтобы избежать возможного шквала оповещений, целесообразно выделять в наборе ключевые (лидирующие) сигналы. Тогда в случае возникновения серьезных проблем или аварий в сети будут формироваться оповещения только по таким наиболее важным сигналам.

Если в наборе, привязанном к целевому объекту, имеются сигналы с признаком **Лидер**, то сначала обрабатываются именно такие лидирующие сигналы, и в зависимости от их статусов выявляется необходимость обработки остальных сигналов набора:

- ❖ если хотя бы один из лидирующих сигналов в наборе имеет «плохой» статус (т.е. отличный от зеленого), то остальные сигналы набора не обрабатываются;

- ❖ если все лидирующие сигналы имеют зеленый статус, то обрабатываются остальные сигналы набора.

Лидирующие сигналы обрабатываются всегда, вне зависимости от статусов других лидирующих сигналов.

2.1.5. Класс оповещений Simple

Класс оповещений – совокупность параметров, определяющих условия, при возникновении которых формируются оповещения.

Логику отправки оповещений определяет тип класса оповещений (параметр класса оповещений). В текущей версии реализован один тип класса оповещений – **Simple**, алгоритм которого предполагает анализ изменения статуса сигнала. В классе оповещений типа **Simple** администратор задаёт:

- ❖ настройки множеств "хороших", "плохих" и "игнорируемых" статусов сигнала;
- ❖ изменения статусов, при которых формируются оповещения. Изменение статуса описывается его переходом между множествами "плохих" и "хороших" статусов: "плохой" -> "плохой", "плохой" -> "хороший", "хороший" -> "плохой", "хороший" -> "хороший";
- ❖ множество сигналов, по которым будут формироваться оповещения (по умолчанию – по всем сигналам);
- ❖ параметры алгоритма формирования повторных оповещений (по умолчанию повторные оповещения не формируются);

Согласно алгоритму необходимость формирования оповещения зависит от принадлежности исходного и нового статуса сигнала множествам "хороших", "плохих" и "игнорируемых" статусов:

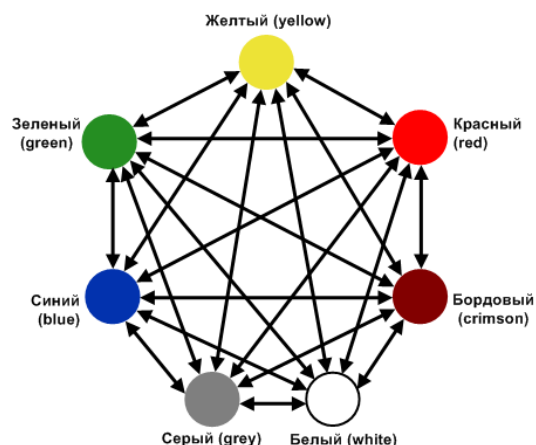
- ❖ оповещение формируется, если изменение статуса удовлетворяет одному из условий, установленных в классе оповещений. Возможные варианты переходов:
 - "плохой" → "хороший": исходный статус входит в множество "плохих" статусов, а новый статус – в множество "хороших" статусов (принадлежность нового статуса множеству "плохих" статусов не рассматривается);
 - "хороший" → "плохой": исходный статус принадлежит множеству "хороших" статусов, а новый статус – множеству "плохих" статусов (принадлежность нового статуса множеству "хороших" статусов не рассматривается);
 - "плохой" → "плохой": оба статуса (исходный и новый) принадлежат множеству "плохих" статусов;
 - "хороший" → "хороший": оба статуса (исходный и новый) принадлежат множеству "хороших" статусов;
 - также можно настроить отправку при любом изменении статусов;
- ❖ оповещение не формируется, если исходный или новый статус сигнала принадлежит множеству "игнорируемых" статусов;
- ❖ после формирования оповещения о переходе сигнала в статус, принадлежащий множеству "плохих" статусов, через заданные промежутки времени отправляются повторные оповещения, если статус сигнала остается "плохим".

Далее в примерах рассмотрены варианты настроек. Для каждого случая на диаграмме справа стрелки показывают переходы между статусами, при которых формируются оповещения.

Пример 1

Оповещения отправляются при любых изменениях статуса сигнала. На практике данный вариант настройки применяется редко. Чтобы настроить такую схему, каждый статус сигнала должен быть отнесен к множеству "хороших" или "плохих" статусов, а множество "игнорируемых" статусов должно быть пустым. Должен быть установлен флаг формирования оповещения при любом изменении статуса.

Статус 'плохой'	<div><div></div><div></div><div></div><div></div></div>
Статус 'игнорировать'	<div><div></div><div></div><div></div><div></div></div>
Статус 'хороший'	<div><div></div><div></div><div></div><div></div></div>
Отправка при любом изменении статуса	on

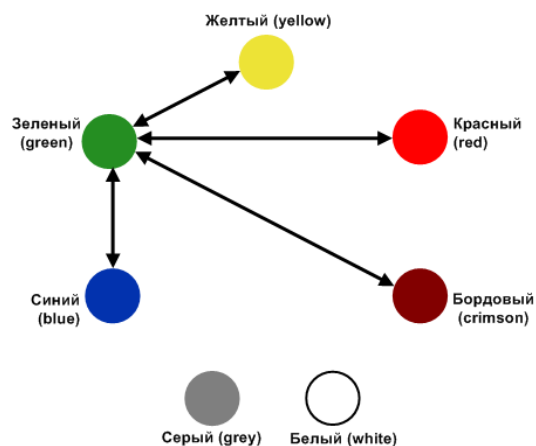


Пример 2

Предустановленные настройки. Оповещения формируются при переходе сигнала из зеленого статуса в любой другой (кроме серого и белого), и обратно: из любого статуса (кроме серого и белого) – в зелёный.

При использовании такой настройки следует учитывать, что оповещения не формируются при переходе статуса из серого в любой "плохой". Например, цепочка последовательных переходов "зеленый → серый → красный" не будет сопровождаться оповещениями, поэтому можно "упустить" переход сигнала в "плохой" статус.

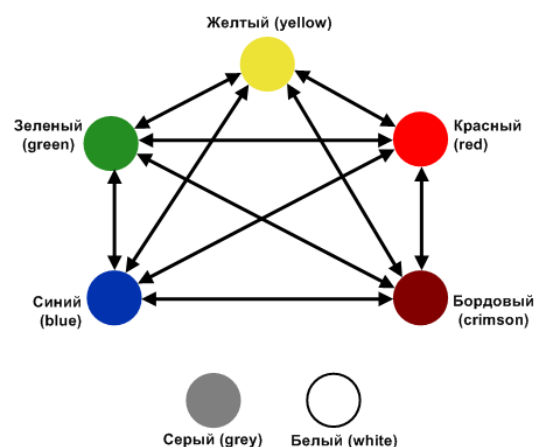
Статус 'плохой'	<div><div></div><div></div><div></div><div></div></div>
Статус 'игнорировать'	<div><div></div><div></div><div></div><div></div></div>
Статус 'хороший'	<div><div></div><div></div><div></div><div></div></div>
Отправка при любом изменении статуса	off
Отправка при переходе хороший-плохой	on
Отправка при переходе хороший-хороший	off
Отправка при переходе плохой-хороший	on
Отправка при переходе плохой-плохой	off



Пример 3




Оповещения формируются при изменениях статуса сигнала, описанных в предыдущем примере, а также при переходе сигнала из одного "плохого" статуса в другой "плохой" статус.

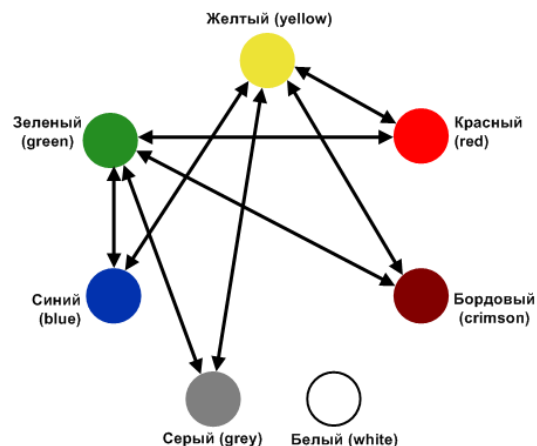
Статус 'плохой'	<div><div></div><div></div><div></div><div></div></div>
Статус 'игнорировать'	<div><div></div><div></div><div></div><div></div></div>
Статус 'хороший'	<div><div></div><div></div><div></div><div></div></div>
Отправка при любом изменении статуса	off
Отправка при переходе хороший-плохой	on
Отправка при переходе хороший-хороший	off
Отправка при переходе плохой-хороший	on
Отправка при переходе плохой-плохой	on



Пример 4

В данном случае "хорошим" статусом, помимо зелёного, является также и жёлтый статус. А серый статус относится к "плохим". Оповещения формируются при переходе из "плохого" статуса в "хороший" (зелёный или жёлтый) и обратно – из "хорошего" в "плохой".

Статус 'плохой'	
Статус 'игнорировать'	
Статус 'хороший'	
Отправка при любом изменении статуса	off
Отправка при переходе хороший-плохой	on
Отправка при переходе хороший-хороший	off
Отправка при переходе плохой-хороший	on
Отправка при переходе плохой-плохой	off



Примечание. Один и тот же набор переходов между статусами, при которых отправляются оповещения, может быть настроен разными комбинациями множеств "хороших" и "плохих" статусов. Присутствие статуса в множестве "плохих" статусов влияет на формирование повторных оповещений.

2.1.6. Методы и типы оповещений

Метод оповещений – совокупность параметров, определяющих способ формирования оповещений. Метод оповещений содержит ключевой параметр **тип оповещения** и параметры, конкретизирующие выполняемое действие, например: тип оповещения «отправка сообщения по электронной почте», адреса получателей сообщений, сетевой адрес почтового сервера.

Метод оповещения задается следующими параметрами:

- ❖ имя метода;
- ❖ тип оповещений;
- ❖ признак активности метода:
 - on** – метод включен, оповещения формируются;
 - off** – метод отключен, оповещения не формируются (можно использовать для временной блокировки отправки оповещений);
- ❖ параметры формирования и доставки оповещения – набор этих параметров определяется типом оповещений.

В приведенном списке первые два параметра имеют только глобальные значения, т.е. они задаются при создании метода оповещений и не подлежат последующему изменению. Для остальных параметров метода оповещения могут быть заданы локальные значения.

Тип оповещения – вид уведомления, например, отправка сообщения по электронной почте или SMS, создание записи в лог-файле или базе данных, передача сообщения о событии в стороннюю систему, выполнение иной программной процедуры (скрипта).

Реализованы следующие типы оповещений:

- ❖ **Email** – отправка письма по электронной почте;
- ❖ **Exec** – выполнение программы/скрипта;
- ❖ **Log** – запись в лог-файле;
- ❖ **MYSQL** – запись в базе данных MYSQL;
- ❖ **SMS** – отправка SMS-сообщения;
- ❖ **SSH** – передача команды на сторонний ресурс по протоколу SSH;
- ❖ **Syslog** – запись в системном журнале;
- ❖ **Trap** – прерывание SNMP;
- ❖ **WsNotify** – уведомление во всплывающем окне в Web-интерфейсе;
- ❖ **ZodiacEvent** – формирование сообщений типа `zodiacEvent` и отправка их в шину ESB.

2.1.7. Универсальный шаблон оповещений

Универсальный шаблон оповещений – совокупность правил, определяющих необходимость формирования оповещений и способы их отправки.

Универсальный шаблон оповещений представляет собой последовательность шагов. Реализованы шаги следующих типов:

- ❖ **StateFilter** — фильтр сигналов в зависимости от изменения состояния;
- ❖ **Backoff** — повторные напоминания;
- ❖ **Delay** — выполнение задержки;
- ❖ **Digest** — группировка сигналов в массив;
- ❖ **Email** — отправка уведомления по электронной почте;
- ❖ **SubalertFilter** — фильтрация сигналов по указанному свойству (полная эквивалентность или регулярное выражение);
- ❖ **Log** — создание записи в лог-файле;
- ❖ **RenderFile** — генерация строковых данных на основе шаблона из файла (используется шаблонизатор [jinja2](#));
- ❖ **RenderString** — генерация строковых данных на основе шаблона из строки (используется шаблонизатор [jinja2](#));
- ❖ **SSH** — выполнение команд через SSH;
- ❖ **SysLog** — запись данных в системный лог;
- ❖ **WsNotify** — формирование уведомления во всплывающих окнах в Web-интерфейсе;
- ❖ **TargetsInfos** — получает перечень сигналов и в том же порядке возвращает информацию о целевых объектах, к которым привязаны сигналы из перечня;
- ❖ **Exec** — выполняет команду на локальном хосте;
- ❖ **RocketChat** — отправляет уведомление в RocketChat.

Администратор привязывает универсальные шаблоны оповещений к целевым объектам, также как и наборы оповещений.

Формирование и отправка оповещений по универсальным шаблонам осуществляется системным процессом `notifng`.

Системный процесс `notifng` осуществляет обработку сигналов (анализирует изменения значений и статусов сигналов) и выполняет шаги, настроенные в универсальных шаблонах. Он использует конвейерный метод – выполняет цепочку преобразований, и в результате отправляет оповещение пользователю одним из доступных способов.

Каждый шаг содержит зависимости (может быть более одного) от других шагов. Для перехода между шагами используется два ключевых слова:

- ❖ имя шага, который завершил выполнение;
- ❖ выход (`output`) – строковое значение для дополнительной идентификации результата выполнения шага. По умолчанию все шаги обладают двумя выходами:
 - `return` – выход по умолчанию, сообщающий о завершении выполнения шага и передающий некоторые данные из шага;
 - `cancel` – выход, сообщающий зависимым шагам о необходимости отмены, окончании выполнения дочернего шага.

Данный список может быть расширен в некоторых шагах, предусматривающих различное поведение в зависимости от свойств сигнала, например в **StateFilter**.

Для активации цепи обработки используется условный шаг `entry` с выходом `return`. Выход `return` не является обязательным и устанавливается по умолчанию при отсутствии иного значения выхода для всех шагов, т.е. запись `[[entry]]` эквивалентна `[[entry/return]]`.

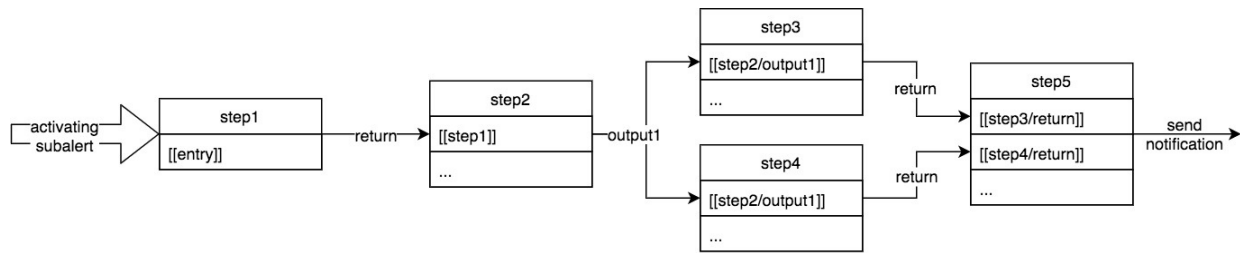


Рисунок 8. Схема выполнения шагов, настроенных в шаблоне, системным процессом notifng

3. НАСТРОЙКА ПМ PHOENIXDS ALARM MANAGEMENT

Исполняемые утилиты ПМ **PhoenixDS Alarm Management** находятся в директории `/usr/local/lancelot/bin`. Далее сделаны следующие предположения:

- ❖ установлено соединение с сервером от имени пользователя `avalon`;
- ❖ указанная выше директория прописана в переменной `PATH`, или пользователь перешел в эту директорию;
- ❖ на сервер установлен ПМ **PhoenixDS Alarm Management**.

В ПМ **PhoenixDS Alarm Management** следует работать под пользователем `avalon`, а не под `root`. Переключение пользователя осуществляется с помощью команды `su - avalon`.

3.1. Настройка СУБД ClickHouse для хранения данных о сигналах

Рекомендуется для хранения информации о состоянии сигналов использовать базу данных ClickHouse (вместо БД MySQL, которая использовалась в предыдущих версиях). СУБД ClickHouse следует устанавливать на отдельный сервер.

Для настройки установленной СУБД ClickHouse необходимо выполнить следующие шаги:

1. установить и настроить службу Apache ZooKeeper, см. [Установка и настройка Apache ZooKeeper](#);
2. настроить кластер ClickHouse, см. [Настройка кластера ClickHouse](#);
3. настроить таблицы БД ClickHouse, см. [Настройка таблиц в БД ClickHouse](#);
4. переключить ПМ **PhoenixDS Alarm Management** на использование БД ClickHouse, см. [Переключение ПМ PhoenixDS Alarm Management](#);
5. перенести данные о состоянии сигналов из БД MySQL в БД ClickHouse.

3.1.1. Установка и настройка Apache ZooKeeper

Apache ZooKeeper – это централизованная служба для поддержки информации о конфигурации, именования, обеспечения распределенной синхронизации и предоставления групповых служб. Чтобы установить на сервер Apache ZooKeeper, следует:

1. скачать архив требуемой версии с официального сайта <http://zookeeper.apache.org/>;
2. распаковать архив в директорию, из которой будет осуществляться запуск Apache ZooKeeper;
3. настроить конфигурационный файл `zoo.cfg`. Чтобы предотвратить неограниченное потребление пространства на диске логами и снапшотами ZooKeeper, рекомендуется в конец файла конфигурации `zoo.cfg` добавить приведенный ниже фрагмент.

Фрагмент файла конфигурации `zoo.cfg`, предлагаемый документацией Clickhouse:

```
# http://hadoop.apache.org/zookeeper/docs/current/zookeeperAdmin.html

# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=30000
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=10

maxClientCnxns=2000

maxSessionTimeout=600000000
# the directory where the snapshot is stored.
dataDir=/opt/zookeeper/data
# Place the dataLogDir to a separate physical disc for better performance
```

```

dataLogDir=/opt/zookeeper/logs

autopurge.snapRetainCount=10
autopurge.purgeInterval=1

# To avoid seeks ZooKeeper allocates space in the transaction log
# file in
# blocks of preAllocSize kilobytes. The default block size is 64M.
# One reason
# for changing the size of the blocks is to reduce the block size
# if snapshots
# are taken more often. (Also, see snapCount).
preAllocSize=131072

# Clients can submit requests faster than ZooKeeper can process
# them,
# especially if there are a lot of clients. To prevent ZooKeeper
# from running
# out of memory due to queued requests, ZooKeeper will throttle
# clients so that
# there is no more than globalOutstandingLimit outstanding requests
# in the
# system. The default limit is 1,000. ZooKeeper logs transactions to a
# transaction log. After snapCount transactions are written to a
# log file a
# snapshot is started and a new transaction log file is started.
# The default
# snapCount is 10,000.
snapCount=3000000

# If this option is defined, requests will be will logged to a
# trace file named
# traceFile.year.month.day.
#traceFile=

```

4. запустить Apache ZooKeeper согласно инструкции https://zookeeper.apache.org/doc/r3.6.1/zookeeperStarted.html#sc_InstallingSingleMode.

Кроме того, необходимо проверить, что на сервере установлены необходимые версии пакетов java:

- ❖ java-1.8.0-openjdk.x86_64;
- ❖ java-1.8.0-openjdk-devel.x86_64.

3.1.2. Настройка кластера ClickHouse

После запуска Apache ZooKeeper необходимо создать кластер с репликацией и связать его с работающим Apache ZooKeeper.

Основные рекомендации по настройке см. в документации Clickhouse <http://clickhouse.tech/docs/ru/engines/table-engines/mergetree-family/replication/>.

Пример

Чтобы настроить распределенную реплицированную систему, состоящую из одного узла с возможностью расширения, необходимо выполнить следующие шаги:

1. в конфигурационный файл config.xml в раздел remote_servers добавить новый узел с именем lancelet_alerts и заполнить его необходимыми параметрами (в примере опущены тестовые узлы из конфигурации по-умолчанию):

```

<remote_servers incl="clickhouse_remote_servers">
  <lancelet_alerts>
    <shard>
      <internal_replication>true</internal_replication>
      <weight>1</weight>
    </shard>
  </lancelet_alerts>
</remote_servers>

```

```

    <replica>
      <host>shard_host</host>
      <port>shard_port</port>
    </replica>
  </shard>
</lancelot_alerts>
</remote_servers>

```

Пример настройки из документации Clickhouse: <http://clickhouse.tech/docs/ru/engines/table-engines/special/distributed/>;

2. для подключения кластера ClickHouse к Apache ZooKeeper в конфигурационном файле `config.xml` заполнить раздел `zookeeper`:

```

<zookeeper>
  <node>
    <host>zookeeper_host</host>
    <port>2181</port>
  </node>
</zookeeper>

```

3. для настройки правила для создания имен репликаций в конфигурационном файле `config.xml` заполнить раздел `macros`:

```

<macros>
  <layer>1</layer>
  <shard>1</shard>
  <replica>replica_table_name_you_may_enter_your_host_name</replica>
</macros>

```

4. добавить правило для работы с JOIN во время SELECT с множества узлов. Для этого добавить настройку по-умолчанию в конфигурационный файл `users.xml` в раздел `yandex.profiles.default`:

```

<distributed_product_mode>local</distributed_product_mode>

```

5. создать пользователя, которым будут пользоваться сервисы (по-умолчанию используются БД `lancelot_alerts` и пользователь `lancelot` с обычным паролем):

```

<!-- Внутри <yandex><users>...</users></yandex> -->
<lancelot>
  <password_sha256_hex>038c710710f7460cc7b6ef6eeb2c2e1d89dc1796b055a28b52b303b942468e47</password_sha256_hex>
  <!-- Или же пароль можно указать в виде plain text (не рекомендуется): -->
  <password>ava$lon</password> -->
  <networks incl="networks" replace="replace">
    <ip>::/0</ip>
  </networks>
  <profile>default</profile>
  <quota>default</quota>
  <allow_databases>
    <database>lancelot_alerts</database>
  </allow_databases>
</lancelot>

```

Если пользователь уже существует, добавить базу `lancelot_alerts` в раздел `allow_databases`.

3.1.3. Настройка таблиц в БД ClickHouse

Создание таблиц

Для создания таблиц следует выполнить скрипт `data/lancelot-alerts-clickhouse-init.sh` с помощью команды:

```
data/lancelot-alerts-clickhouse-init.sh
```

Необходимые параметры доступа к БД ClickHouse можно задать через конфигурационный файл:

```
./clickhouse-client.xml;
```



```
~/clickhouse-client/config.xml;
/etc/clickhouse-client/config.xml.
```

Пример конфигурационного файла `./clickhouse-client.xml`:

```
<config>
  <user>username</user>
  <password>password</password>
  <secure>False</secure>
</config>
```

Репликация

Для создания таблиц следует выполнить скрипт `data/lancelot-alerts-clickhouse-init.sh` с помощью команды:

```
data/lancelot-alerts-clickhouse-init.sh -r
```

Репликация работает на уровне отдельных таблиц, а не всего сервера. То есть, на сервере могут быть расположены одновременно реплицируемые и не реплицируемые таблицы.

Репликация не зависит от шардирования. На каждом шарде репликация работает независимо.

Реплицируются сжатые данные запросов INSERT, ALTER (см. подробности в описании запроса [ALTER](#)).

Запросы CREATE, DROP, ATTACH, DETACH и RENAME выполняются на одном сервере и не реплицируются.

Удаление старых событий

Если стоит задача не хранить события дольше какого-то интервала, можно воспользоваться механизмом TTL ClickHouse.

Например, если необходимо, чтобы события автоматически удалялись через 3 месяца, можно выполнить следующий запрос:

Образец установки TTL:

```
ALTER TABLE `lancelot_alerts`.`alerts_history` ON CLUSTER `lancelot_alerts`
MODIFY TTL `time_state` + INTERVAL 3 MONTH
```

3.1.4. Переключение ПМ PhoenixDS Alarm Management

Чтобы перевести ПМ **PhoenixDS Alarm Management** на хранение информации о состоянии сигналов в БД ClickHouse, следует:

1. выполнить скрипт:

```
util/configAlertsBackend.py --backend clickhouse
```

С помощью данного скрипта можно также исправить конфигурацию `dataserver` (п. 2). В этом случае его следует вызвать командой:

```
util/configAlertsBackend.py --backend clickhouse --fix-dataserver-config
/usr/local/avalon/etc/dataServer.ini
```

Скрипт сохранит резервную копию файла конфигурации, закомментирует системные процессы `alert` и исправит нумерацию оставшихся;

2. выключить обработку `alert`-сообщений в `/usr/local/avalon/etc/dataServer.ini` (если это не было сделано в пункте 1). Для этого в файле `/usr/local/avalon/etc/dataServer.ini` закомментировать или удалить приведенные ниже секции и исправить нумерацию оставшихся секций `[SERVICES_<N>]`;

```
[SERVICES_4]
name = DS_ALERT_1
thread-pool-size = 3
storage = ALERT
type1 = alertCreate
type2 = alertRemove
```

```
type3 = alertUpdate
type4 = alertUpdateState
type5 = wsNotificationAlert
type6 = wsNotificationAlertDismiss
type7 = wsNotificationAlertDismissAll
```

```
[SERVICES_5]
name = DS_ALERT_2
thread-pool-size = 3
storage = ALERT
type1 = alertHistory
type2 = alertLast
type3 = alertQuery
```

3. сконфигурировать и включить системные процессы `ds-alert`, `ds-alertlrd` с помощью команд:

```
overlord.pl enable ds-alertlrd
overlord.pl enable ds-alert
```

4. перезапустить **PhoenixDS Alarm Management** с помощью команды:

```
systemctl restart phoenixdsd
```

3.1.5. Миграция данных из БД MySQL в БД ClickHouse

Перенос данных о состоянии сигналов осуществляется с помощью утилиты `alerts_mover.py`.

Общий порядок выполнения миграции данных:

1. получить необходимые `UUID` из дерева конфигурации;
2. дополнить таблицу `alerts_info` в БД MySQL соответствующими `UUID`;
3. создать таблицы с движком MySQL в базе данных `mysql` в системе ClickHouse аналогичные тем, что находятся в базе данных `lancelot_alerts` в системе MySQL;
4. с помощью утилиты `alerts_mover.py` перенести данные из БД MySQL в БД ClickHouse согласно выбранным частям (параметр `parts`);
5. удалить временную базу данных из системы ClickHouse;
6. удалить колонку с `UUID` из таблицы `alert_info` в БД MySQL.

4. АДМИНИСТРИРОВАНИЕ ПМ PHOENIXDS ALARM MANAGEMENT

Web-интерфейс позволяет конфигурировать объекты в дереве конфигурации, а также осуществлять настройку и администрирование ПМ PhoenixDS Alarm Management.

Для администратора (пользователя root) на иконке в заголовке Web-страницы отображается буква **A**.

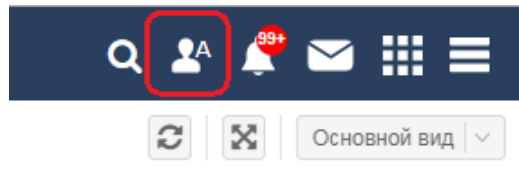



Рисунок 9. Иконка администратора в заголовке Web-интерфейса

Меню администратора (пользователя root), помимо стандартных пунктов, содержит пункты, не доступные другим пользователям:

- ❖ **Инструменты --> Управление Integrity** – проверка целостности дерева конфигурации и исправление нарушений;
- ❖ **Обновить лицензию** – актуализация доступных в Web-интерфейсе данных о лицензионных ограничениях в соответствии с установленным файлом лицензии. Информацию об установленной лицензии и текущие значения счетчиков по параметрам, ограниченным лицензионным соглашением, могут просмотреть все пользователи в пункте меню **Информация о лицензии**.

4.1. Редактор сигналов и оповещений (обзор web-приложения)

Web-приложение **Редактор сигналов и оповещений** позволяет настраивать наборы сигналов и наборы оповещений. В настоящем разделе руководства приведено общее описание основного окна редактора. Подробности о настройке сигналов и оповещений см. в разделах: [Настройка сигналов](#) и [Настройка оповещений](#).

Для вызова **Редактора сигналов и оповещений** следует в заголовке Web-интерфейса нажать кнопку  и в меню приложений нажать иконку с названием web-приложения.

Примечание. Иконка доступна, если пользователь входит в группу пользователей с установленной глобальной политикой **Настройка сигналов и оповещений**.

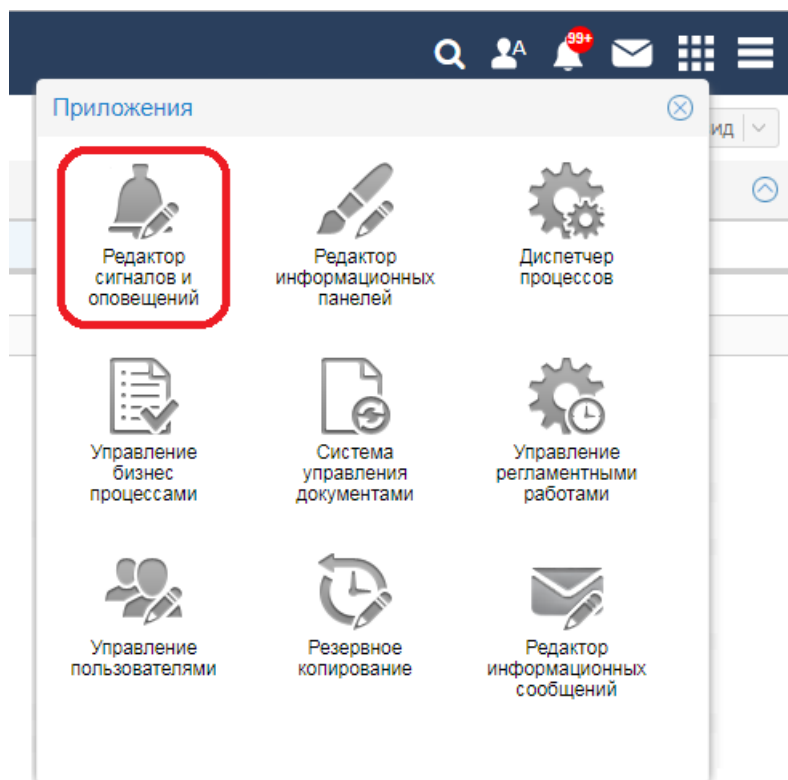


Рисунок 10. Вызов окна Редактор шаблонов сигналов

Основное окно web-приложения имеет следующую структуру:

- ❖ под заголовком окна размещена [панель инструментов](#);
- ❖ слева находится окно **Шаблоны** с вкладками [Сигналы](#) и [Оповещения](#);
- ❖ справа располагается рабочая область, в которой выводится информация о наборе, выбранном на вкладке **Сигналы** или **Оповещения**.

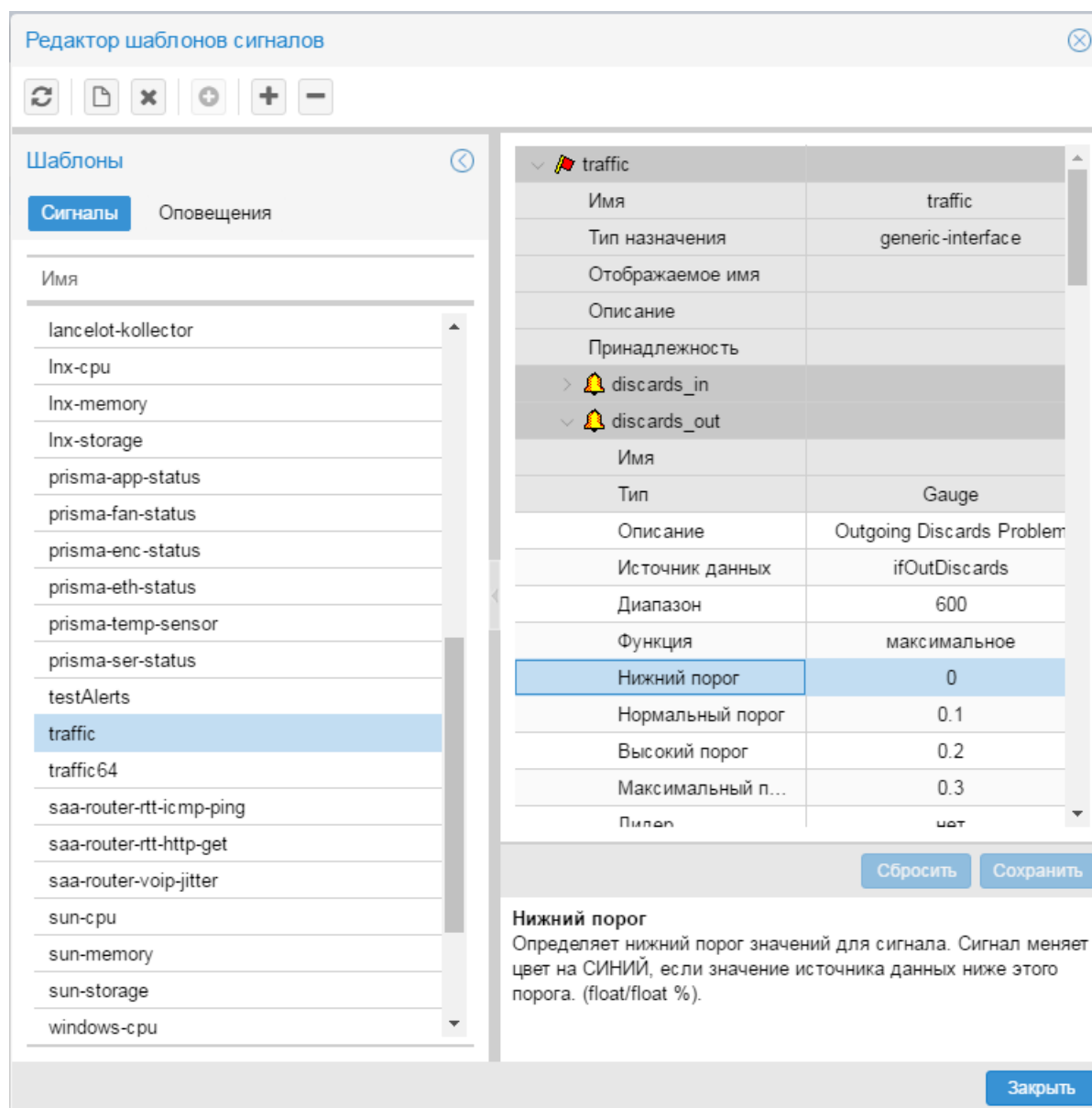



Рисунок 11. Редактор шаблонов сигналов и оповещений

Для выхода из окна **Редактор шаблонов сигналов** можно использовать кнопку  в заголовке окна или кнопку **Закрыть** в правом нижнем углу окна.

4.1.1. Панель инструментов и контекстное меню

В окне **Редактор шаблонов сигналов** для выполнения действий предусмотрены кнопки на панели инструментов, комбинации клавиш и пункты контекстного меню окна **Шаблоны**. Некоторые действия можно выполнить несколькими способами. Чтобы вызвать контекстное меню, следует в окне **Шаблоны** поместить курсор на строку списка на вкладках **Сигналы** или **Оповещения** и кликнуть правой кнопкой мыши.

Ниже перечислены кнопки панели инструментов и соответствующие им пункты контекстного меню и клавиши.



Обновление информации в окне **Редактор сигналов и оповещений** в соответствии с данными сервера PhoenixDS.



| **Новый шаблон...** | **Ins**

Создание нового набора сигналов или набора оповещений.

**| Удалить шаблон | Del**

Удаление выбранного набора сигналов или набора оповещений.

**| Добавить класс оповещения**

Добавление класса оповещения в набор оповещений.

**| Добавить подсигнал / Добавить метод**

Добавление сигнала / метода оповещений в выбранный набор сигналов / набор оповещений.

**| Удалить подсигнал / Удалить метод**

Удаление выбранного сигнала / метода из выбранного набора сигналов / набора оповещений.

Копировать шаблон...

Создание копии выбранного набора сигналов / набора оповещений.

Переименовать шаблон... | F2

Изменение отображаемого имени выбранного набора сигналов / набора оповещений.

4.1.2. Вкладка Сигналы

Вкладка **Сигналы** предназначена для настройки наборов сигналов и содержит список существующих наборов сигналов, как предустановленных, так и созданных администратором.

Для набора сигналов, выбранного на вкладке **Сигналы**, в рабочей области справа формируется двухуровневая иерархия раскрываемых блоков с параметрами.

Головной блок содержит параметры набора сигналов:

- ❖ **Имя** – наименование набора сигналов;
- ❖ **Тип назначения** – тип целевого объекта, с которым связан набор сигналов;
- ❖ **Отображаемое имя** – имя, под которым набор сигналов отображается в Web-интерфейсе, в том числе и в списке на вкладке **Сигналы**. Если параметр **Отображаемое имя** не задан, то на Web-страницах выводится значение параметра **Имя**.
- ❖ **Описание** – описание набора сигналов (комментарий);
- ❖ **Принадлежность** – имя сервиса, отвечающего за обработку сигналов. По умолчанию используется предустановленный сервис [alerts](#).

Вложенные блоки соответствуют сигналам в составе набора. Для раскрытия и сворачивания этих блоков можно использовать кнопки со стрелками (слева от пиктограммы). Каждый блок содержит:

- ❖ общие параметры сигнала, не подлежащие редактированию (на затененном фоне);
- ❖ глобальные значения параметров.

В поле под иерархией блоков выводится описание параметра, выбранного в данный момент.

Если пользователь изменил значение параметра, но еще не сохранил его, то соответствующая ячейка помечается красным треугольником в левом верхнем углу, а в нижней части рабочей области активизируются кнопки:

- ❖ **Сбросить** – отказ от внесенных изменений и возврат к прежним значениям;
- ❖ **Сохранить** – сохранение изменений на сервере PhoenixDS.

Действие обеих кнопок распространяется сразу на все измененные параметры.

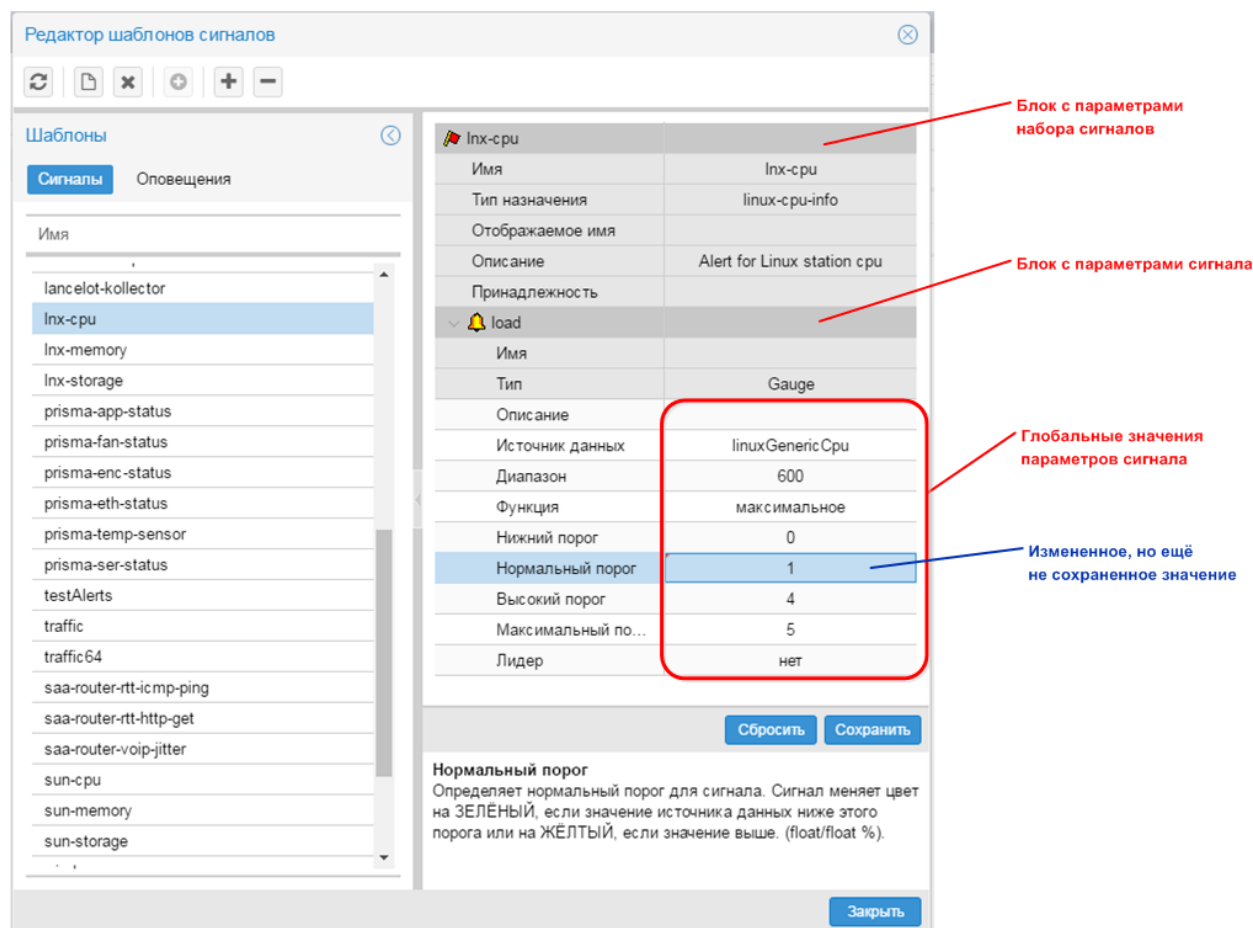


Рисунок 12. Рабочая область вкладки Сигналы

Администратор (пользователь `root`) имеет следующие возможности:

- ❖ [создать новый набор сигналов](#);
- ❖ [добавить сигнал в набор сигналов](#);
- ❖ [удалить сигнал из набора сигналов](#);
- ❖ [скопировать набор сигналов](#);
- ❖ [переименовать набор сигналов](#);
- ❖ [изменить глобальные значения параметров сигнала](#);
- ❖ [удалить набор сигналов](#).

4.1.3. Вкладка Оповещения


Вкладка **Оповещения** предназначено для настройки наборов оповещений и содержит список существующих наборов оповещений, как предустановленных, так и созданных администратором.

Для набора оповещений, выбранного на вкладке **Оповещения**, в рабочей области справа формируется двухуровневая иерархия раскрываемых блоков с параметрами.

Головной блок с пиктограммой  содержит параметры набора оповещений:

- ❖ **Имя** – наименование набора оповещений;
- ❖ **Описание** – описание набора оповещений (комментарий);
- ❖ **Отображаемое имя** – имя, под которым набор сигналов отображается в Web-интерфейсе.

Вложенные блоки соответствуют классу оповещений и методам оповещений в составе набора.

Блок класса оповещений обозначен пиктограммой , блоки методов оповещений – пиктограммами, соответствующими типу оповещения. Для раскрытия и сворачивания этих блоков можно использовать кнопки со стрелками (слева от пиктограммы). Каждый блок содержит:

- ❖ общие параметры, не подлежащие редактированию;
- ❖ глобальные значения параметров.

В поле под иерархией блоков выводится описание параметра, выбранного в данный момент.

Параметры, недоступные для редактирования, отображаются на затененном фоне.

Если администратор изменил значение параметра, то соответствующая ячейка в левом верхнем углу помечается красным треугольником, а в нижней части рабочей области активизируются кнопки:

- ❖ **Сбросить** – отказ от внесенных изменений и возврат к прежним значениям;
- ❖ **Сохранить** – сохранение изменений на сервере PhoenixDS.

Действие обеих кнопок распространяется сразу на все измененные параметры.

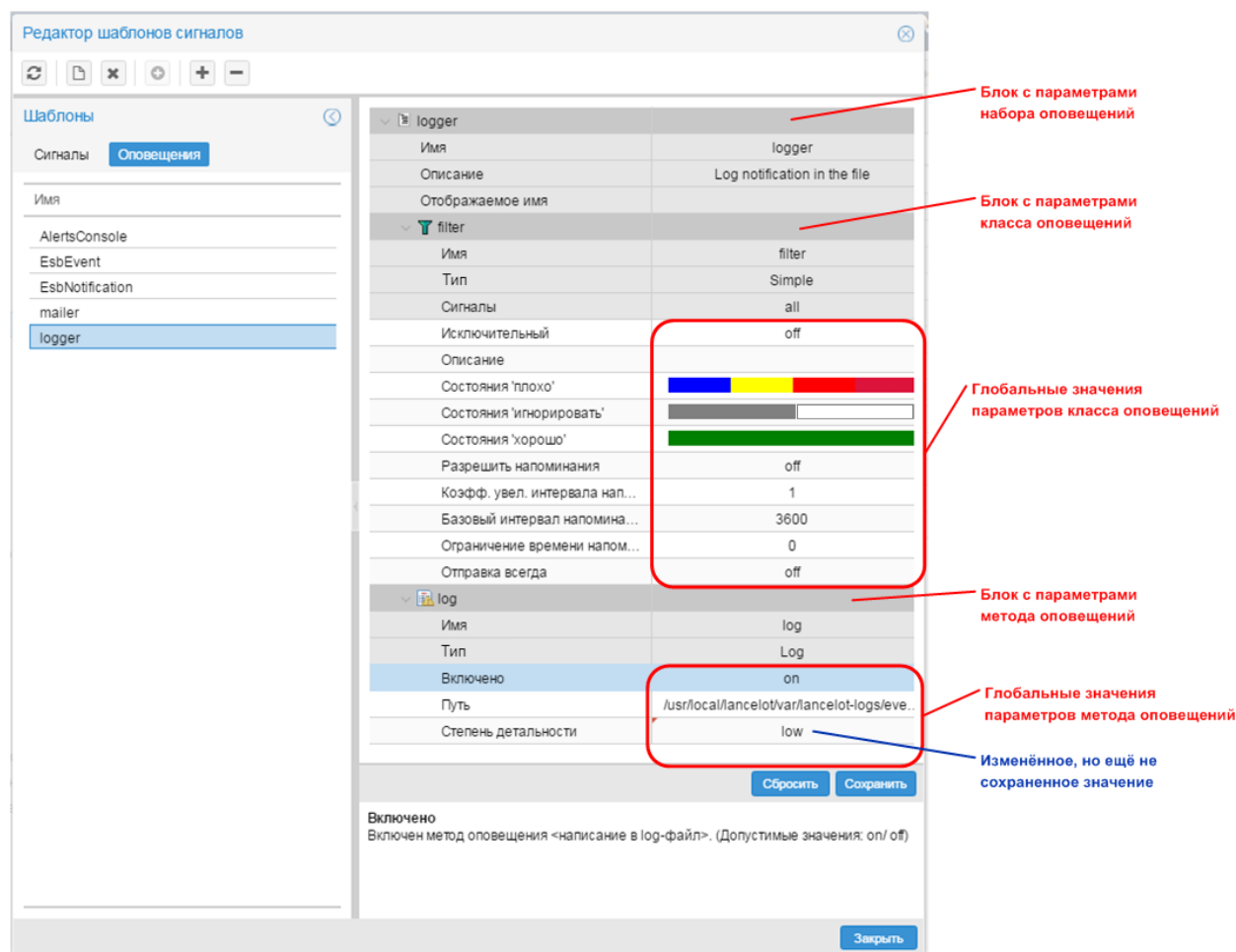


Рисунок 13. Рабочая область окна Оповещения

Администратор (пользователь root) имеет следующие возможности:

- ❖ [создать новый набор оповещений и добавить в него класс оповещений](#);
- ❖ [добавить метод оповещений в набор оповещений](#);
- ❖ [удалить метод/класс оповещений из набора оповещений](#);
- ❖ [скопировать набор оповещений](#);
- ❖ [переименовать набор оповещений](#);
- ❖ [изменить глобальные значения параметров метода / класса оповещений](#);
- ❖ [удалить набор оповещений](#).

4.2. Настройка сигналов

Процесс настройки сигналов включает в себя два этапа:

- ❖ настройка наборов сигналов в Редакторе сигналов и оповещений выполняется пользователями, входящими в группу с установленной глобальной политикой **Настройка сигналов и оповещений**;

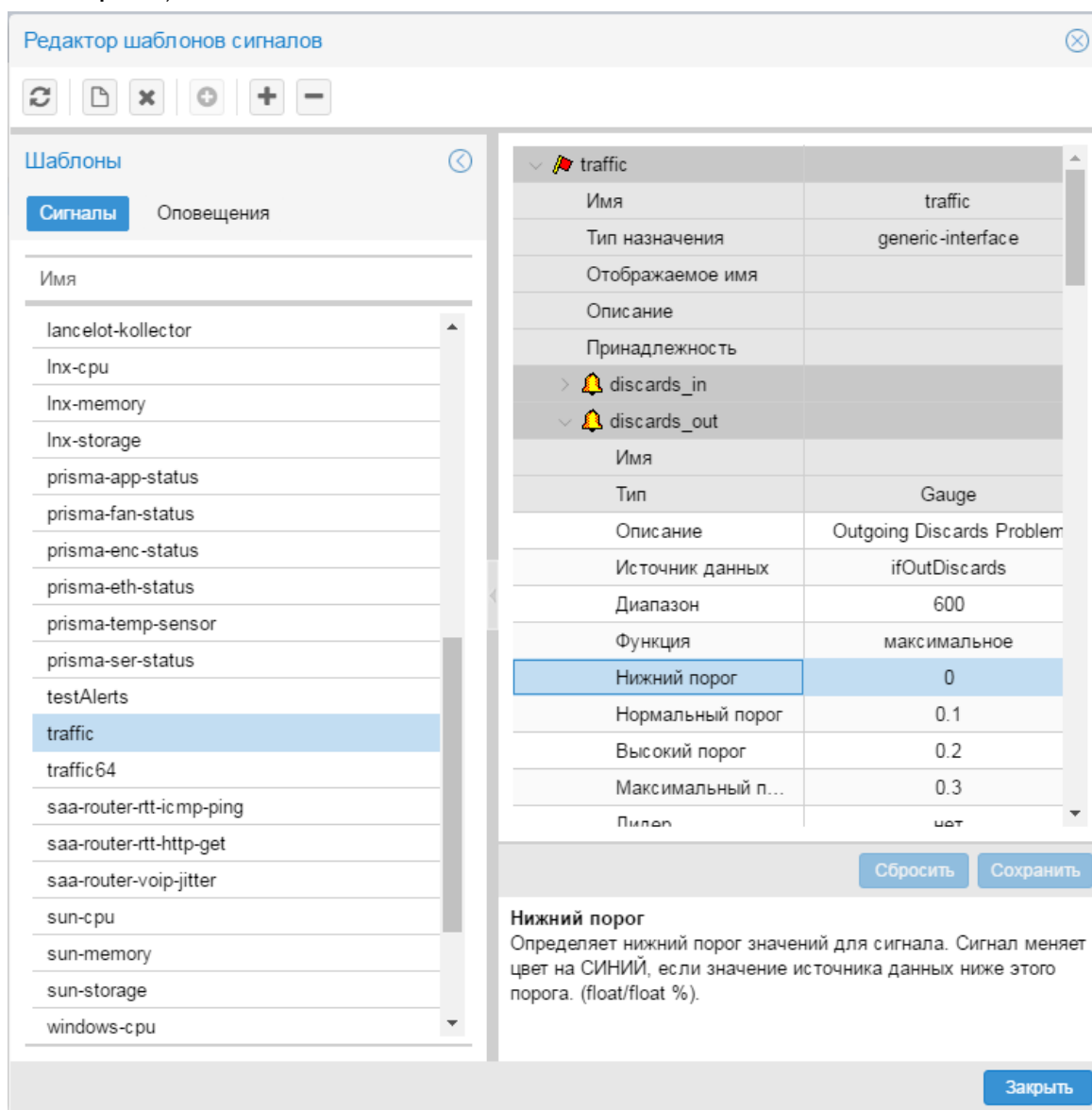


Рисунок 14. Редактор сигналов и оповещений

- ❖ [назначение наборов сигналов целевым объектам](#) – выполняется через контекстное меню окна **Дерево конфигурации**.

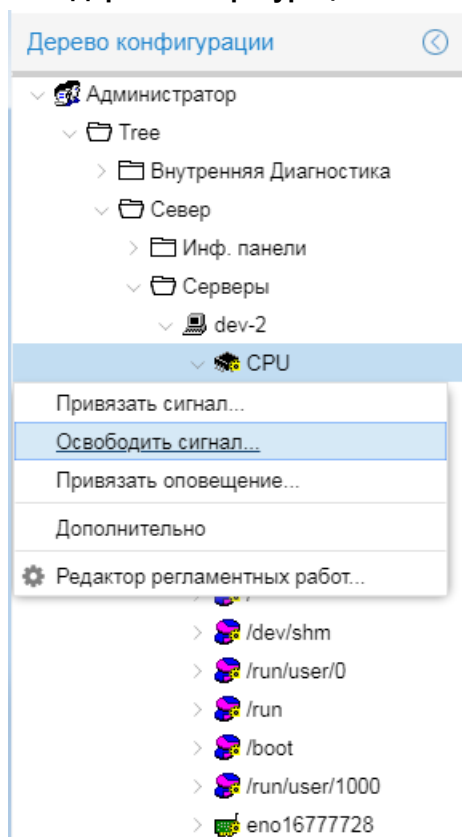



Рисунок 15. Привязка наборов сигналов к целевым объектам

При настройке сигналов следует руководствоваться правилами:

- ❖ набор сигналов может содержать один или несколько сигналов;
- ❖ к целевому объекту можно прикрепить только один набор сигналов.

4.2.1. Настройка наборов сигналов

Настройка наборов сигналов осуществляется в окне **Редактор шаблонов сигналов**, которое вызывается из меню приложений (кнопка  в заголовке Web-интерфейса).

Чтобы настроить новый набор сигналов, администратору следует:

- ❖ сначала [создать новый набор сигналов](#);
- ❖ затем [добавить в него сигналы](#).


Для ускорения настройки можно [скопировать](#) существующий набор сигналов и внести в него необходимые изменения.

Администратор также имеет следующие возможности:

- ❖ [удалить сигнал из набора сигналов](#);
- ❖ [переименовать набор сигналов](#);
- ❖ [изменить глобальные значения параметров сигнала](#);
- ❖ [удалить набор сигналов](#).

4.2.1.1. Создание нового набора сигналов

Чтобы создать новый набор сигналов, следует:

1. в окне **Шаблоны** открыть вкладку **Сигналы** и в контекстном меню выбрать пункт **Новый шаблон**, либо нажать клавишу **Ins** или кнопку  на панели инструментов;

2. в появившемся окне в поле **Имя** ввести имя нового набора сигналов;

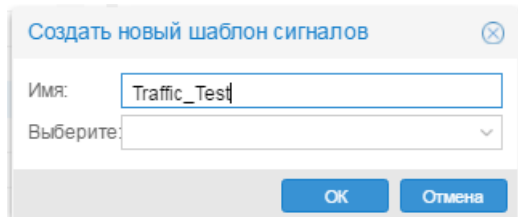



Рисунок 16. Ввод имени нового набора сигналов

3. в поле **Выберите** следует указать тип целевого объекта, к которому будет применяться создаваемый набор; для этого нажать кнопку . Откроется окно **Выбор элемента** с деревом конфигурации;

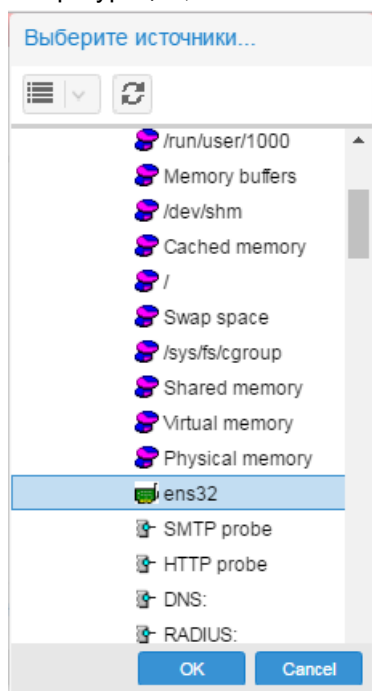


Рисунок 17. Выбор типа целевого объекта

4. выбрать целевой объект нужного типа и нажать **ОК**.

В результате созданный набор сигналов будет добавлен в список вкладки **Сигналы**. В рабочей области будут заполнены поля **Имя шаблона** и **Тип назначения**. При необходимости следует заполнить поле:

Принадлежность

Имя (или шаблон имени) системного процесса, отвечающего за обработку сигналов. По умолчанию сигналы обрабатывает предустановленный системный процесс [alerts](#) и заполнять поле **Принадлежность** не нужно.

Однако, для распределения нагрузки в ПМ **PhoenixDS Alarm Management** может быть запущено несколько системных процессов, созданных на основе процесса `alerts.pl`. Если поле **Принадлежность** не заполнено, то обрабатывать сигналы данного набора могут системные процессы с именами вида `alertsN`, где N – необязательный порядковый номер. Например, системный процесс `alerts` или `alerts42` может обработать сигналы, а системный процесс `alerts-mega` – не сможет.

Аналогичный принцип соблюдается, если поле **Принадлежность** заполнено. Например, если в нем указан системный процесс `super_alerts`, то сигналы могут быть обработаны системными процессами с именами `super_alerts` и `super_alerts98`, и не могут быть обработаны системными процессами `alerts` и `alerts_super`.

Такую жесткую привязку набора сигналов к системному процессу с именем (шаблоном имени), указанным в поле **Принадлежность**, можно отключить, если для системного процесса на основе процесса `alerts.pl` установить параметр `ignore-belongs-to` (но обычно в этом нет необходимости).

См. также [Распределение нагрузки между системными процессами alerts для оптимизации производительности ПП PhoenixDS](#).

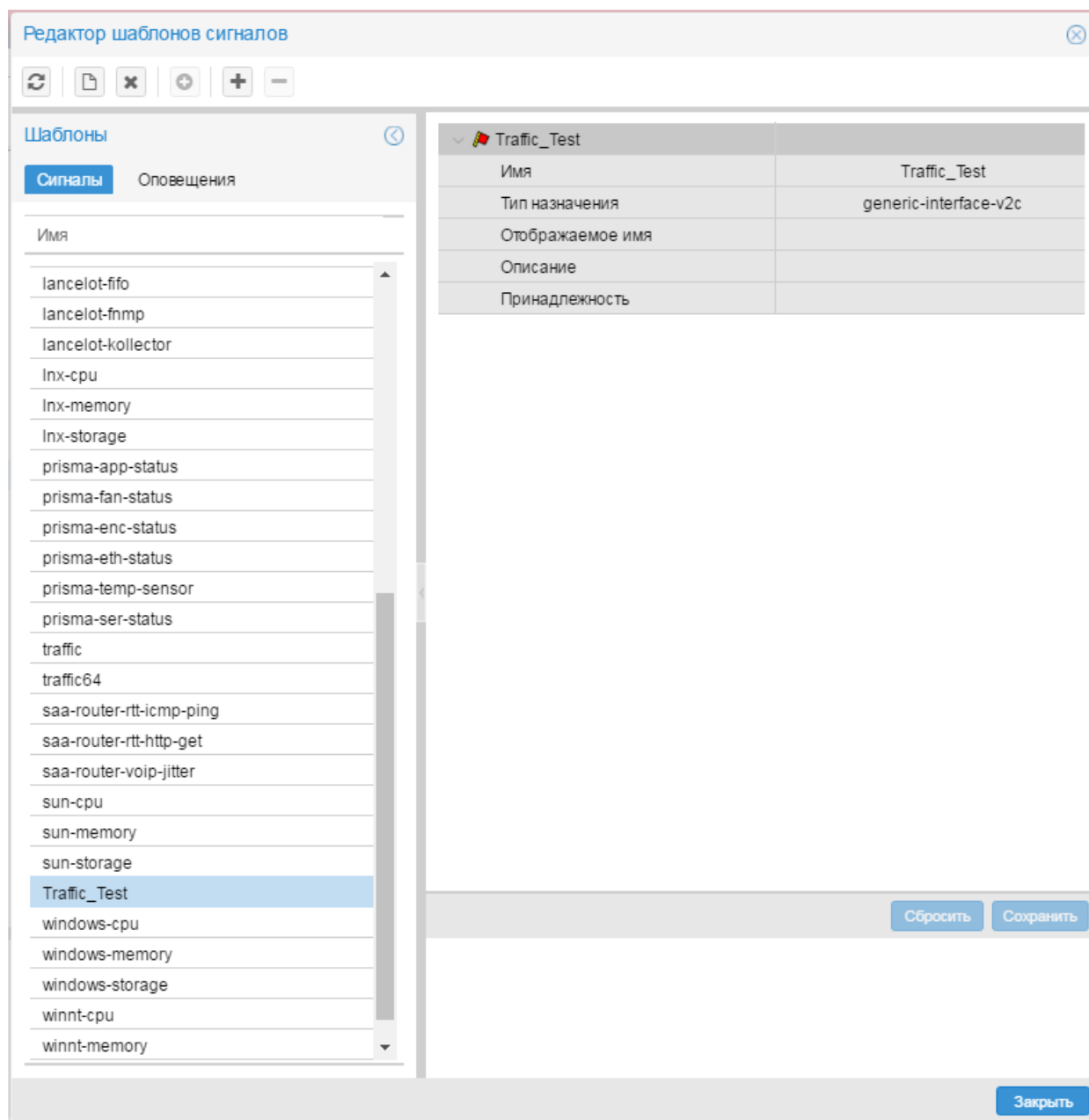


Рисунок 18. Новый набор сигналов в списке

Далее в созданный набор следует [добавить сигналы](#).

4.2.1.2. Добавление сигнала в набор сигналов

Для добавления сигнала в набор следует:

1. в окне **Сигналы** поместить курсор на набор сигналов, к которому нужно добавить сигнал;
2. выполнить одно из действий:

- ❖ в контекстном меню выбрать пункт **Добавить подсигнал** и далее – подпункт, соответствующий типу сигнала;

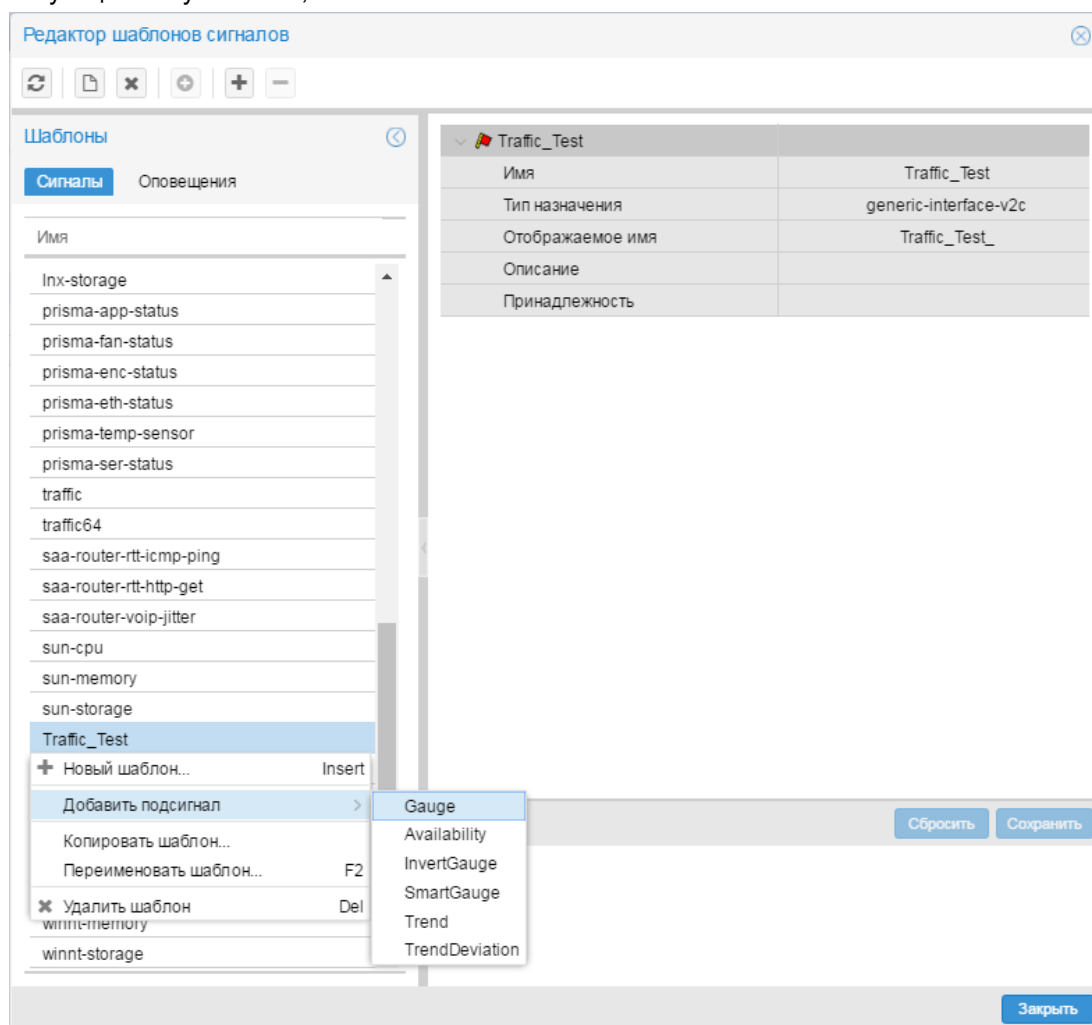


Рисунок 19. Меню для добавления сигнала в набор

- ❖ на панели инструментов нажать кнопку  и в появившемся окне в списке **Подсигнал** выбрать тип сигнала;

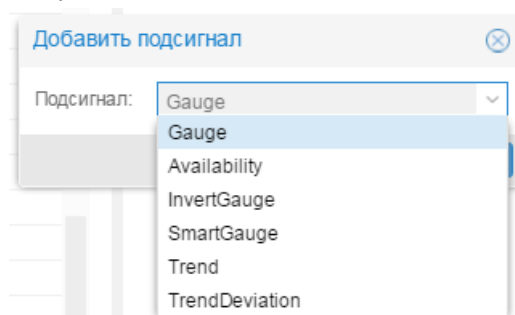


Рисунок 20. Выбор типа сигнала

3. в появившемся окне в поле **Имя подсигнала** ввести имя сигнала:

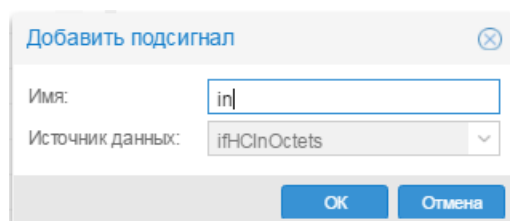


Рисунок 21. Ввод имени сигнала

4. в списке **Источник данных** указать источник данных. Рекомендуется проверить доступные источники данных в MIB устройства;
5. нажать кнопку **ОК**. В результате в рабочую область справа будет добавлен блок с параметрами сигнала;
6. в рабочей области ввести глобальные значения параметров сигнала или оставить значения по умолчанию. Поля формируются в соответствии с типом сигнала. Серые поля недоступны для редактирования. Описание полей см. в справочнике Типы сигналов;;
7. после заполнения полей рабочей области нажать кнопку **Сохранить**.

Редактор шаблонов сигналов

Шаблоны

Сигналы Оповещения

Имя

- Inx-storage
- prisma-app-status
- prisma-fan-status
- prisma-enc-status
- prisma-eth-status
- prisma-temp-sensor
- prisma-ser-status
- traffic
- traffic64
- saa-router-rtt-icmp-ping
- saa-router-rtt-http-get
- saa-router-voip-jitter
- sun-cpu
- sun-memory
- sun-storage
- Traffic_Test_**
- win_disk_storage
- windows-cpu
- windows-memory
- windows-storage
- winnt-cpu
- winnt-memory
- winnt-storage

Traffic_Test	
Имя	Traffic_Test
Тип назначения	generic-interface-v2c
Отображаемое имя	Traffic_Test_
Описание	
Принадлежность	
in	
Имя	
Тип	Gauge
Описание	
Источник данных	ifHCInOctets
Диапазон	600
Функция	максимальное
Нижний порог	0
Нормальный порог	55%
Высокий порог	80%
Максимальный порог	
Лидер	нет

Сбросить Сохранить

Нормальный порог
 Определяет нормальный порог для сигнала. Сигнал меняет цвет на ЗЕЛЁНЫЙ, если значение источника данных ниже этого порога или на ЖЁЛТЫЙ, если значение выше. (float/float %).

Закрыть

Рисунок 22. Сигнал добавлен в набор сигналов

4.2.1.3. Удаление сигнала из набора сигналов

Для удаления сигнала из набора сигналов следует:

1. на вкладке **Сигналы** поместить курсор на набор сигналов, из которого требуется удалить сигнал;
2. выполнить одно из действий:
 - ❖ в контекстном меню выбрать пункт **Удалить подсигнал** и далее – подпункт с наименованием сигнала;

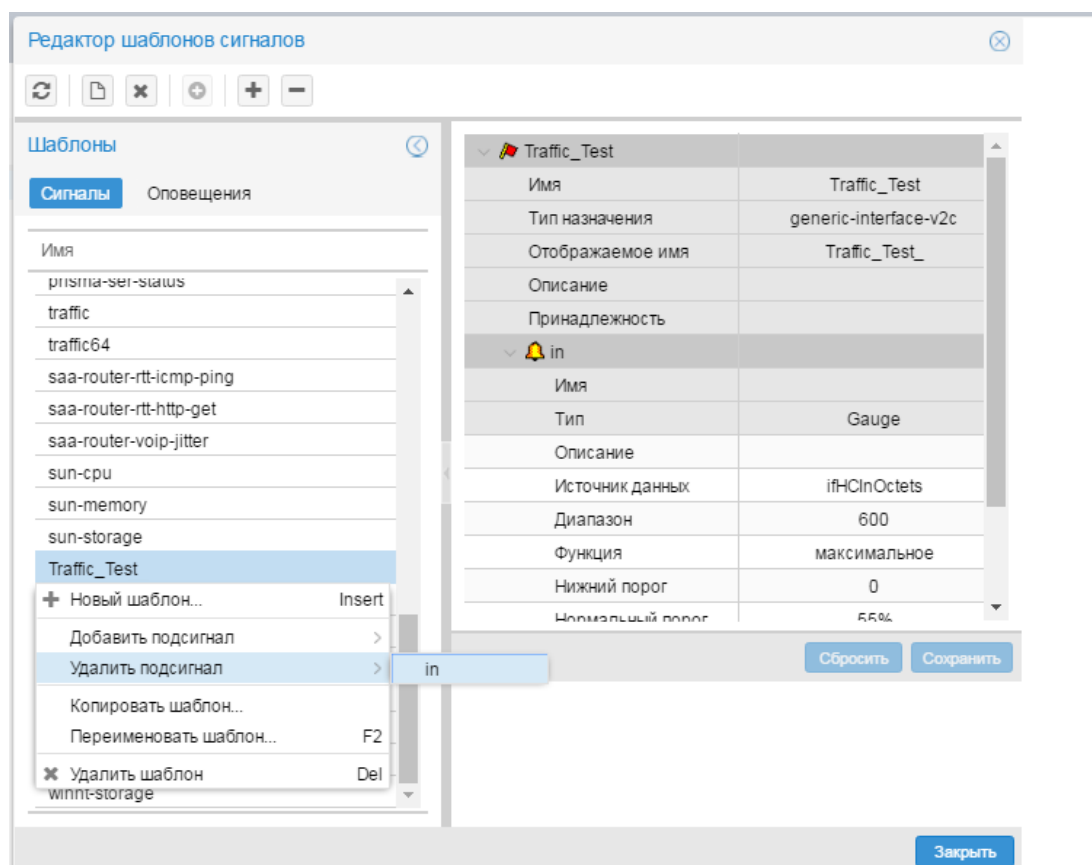



Рисунок 23. Меню для удаления сигнала из набора

- ❖ на панели инструментов нажать кнопку  и в появившемся окне в списке **Подсигнал** выбрать сигнал;

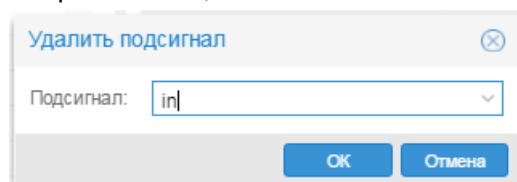


Рисунок 24. Выбор сигнала для удаления

3. в появившемся окне для подтверждения удаления нажать кнопку **Да**, для отказа – кнопку **Нет**.

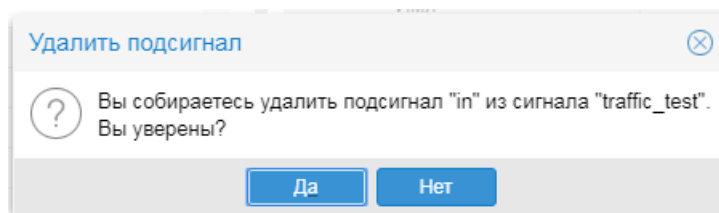


Рисунок 25. Запрос на подтверждение удаления

После подтверждения блок с параметрами сигнала будет удален из рабочей области.

4.2.1.4. Копирование набора сигналов

Для ускорения и упрощения настройки нового набора сигналов можно создать копию существующего набора сигналов и затем внести в него изменения.

Чтобы создать копию набора сигналов, следует:

1. в окне **Сигналы** поместить курсор на набор сигналов, который нужно скопировать;
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Копировать шаблон**;
3. в появившемся окне ввести имя нового набора сигналов; по умолчанию предлагается **<имя исходного набора сигналов>_copy**;

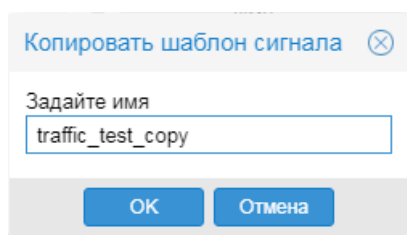


Рисунок 26. Ввод имени нового набора сигналов

4. нажать кнопку **ОК**.

В результате будет создан новый набор сигналов, идентичный исходному набору сигналов.

Для дальнейшей настройки набора сигналов можно выполнить следующие действия:

- ❖ [Переименование набора сигналов](#)
- ❖ [Добавление сигнала в набор сигналов](#)
- ❖ [Изменение глобальных значений параметров сигнала](#)

4.2.1.5. Переименование набора сигналов

Переименование набора сигналов – это изменение его отображаемого имени.

Чтобы переименовать набор сигналов, следует:

1. на вкладке **Сигналы** поместить курсор на нужный набор сигналов;

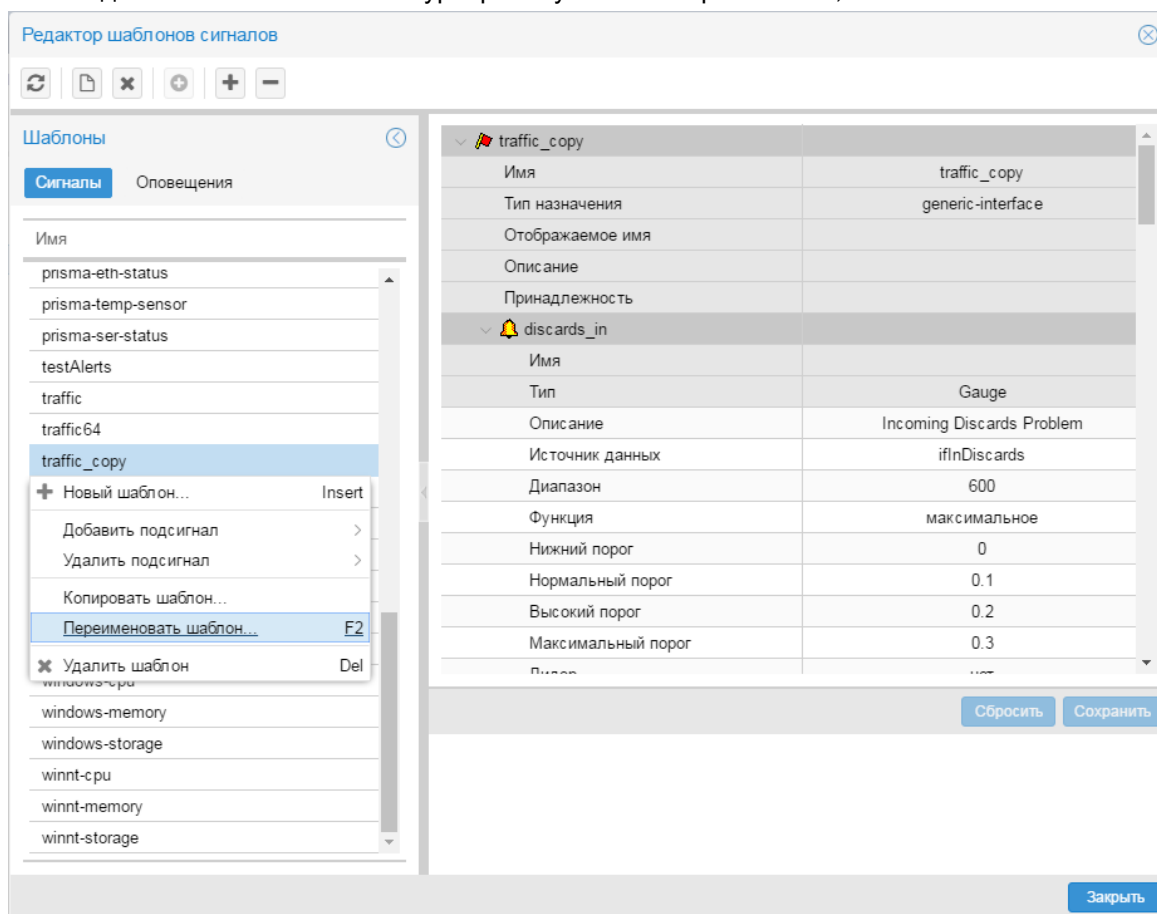


Рисунок 27. Переименованный набор сигналов

2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Переименовать шаблон** (или нажать клавишу **F2**);
3. в появившемся окне ввести новое название набора сигналов;

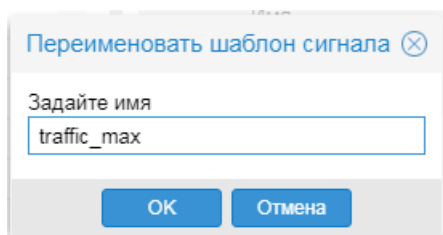


Рисунок 28. Ввод нового отображаемого имени для набора сигналов

4. нажать кнопку **ОК**.

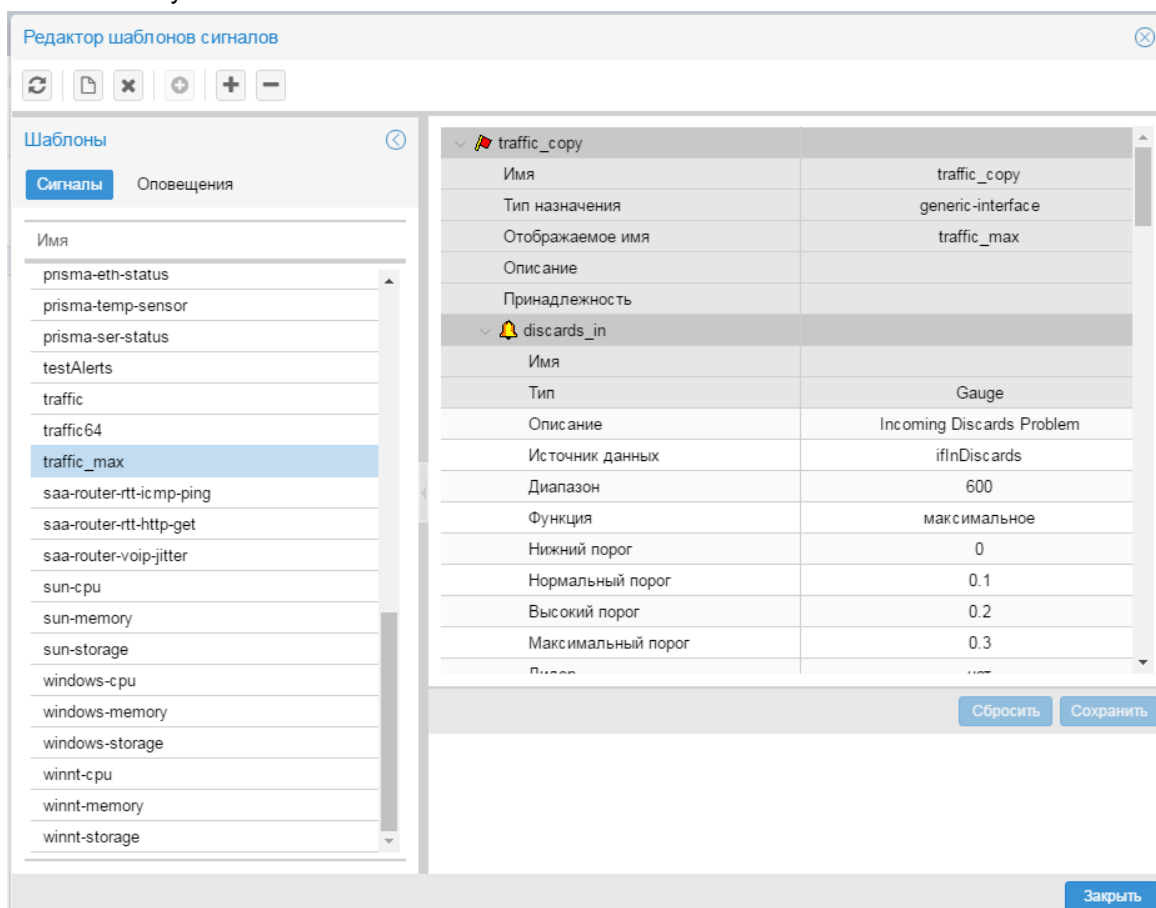


Рисунок 29. Набор сигналов переименован

4.2.1.6. Изменение глобальных значений параметров сигнала

Измененное глобальное значение параметра сигнала распространяется на целевые объекты, к которым сигнал (в составе набора) привязан без указания локального значения.

Для изменения глобальных значений параметров сигнала следует:

1. в окне **Сигналы** поместить курсор на нужный набор сигналов;
2. в рабочей области раскрыть блок с параметрами нужного сигнала и изменить значения полей во втором столбце. Поля формируются в соответствии с типом сигнала. Поля на затененном фоне недоступны для редактирования. Описание полей см. в справочнике Типы сигналов;;
3. для сохранения изменений нажать кнопку **Сохранить**, для отказа от изменений – кнопку **Сбросить**.

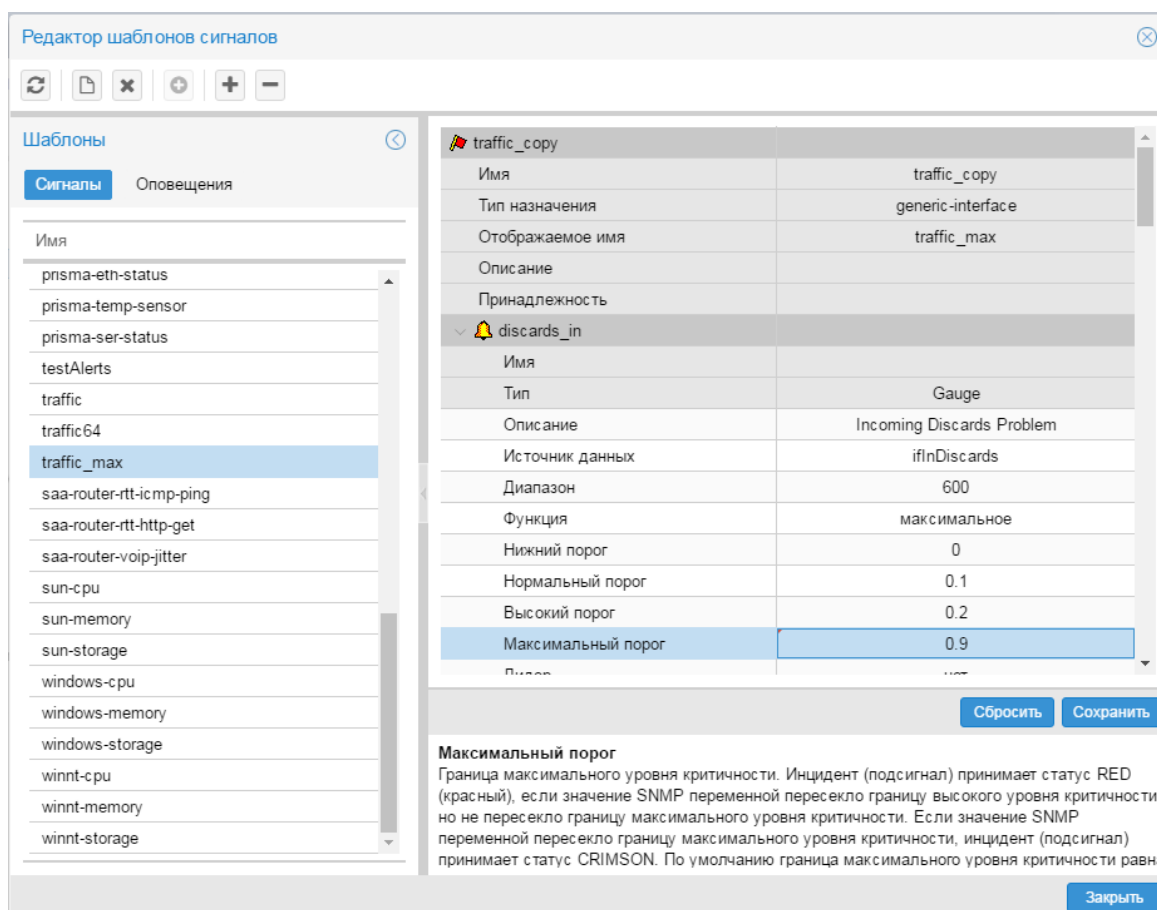



Рисунок 30. Изменение глобальных значений параметров сигнала

4.2.1.7. Удаление набора сигналов

Допускается удаление наборов сигналов, не прикрепленных к целевым объектам.

Чтобы удалить набор сигналов, следует:

1. на вкладке **Сигналы** поместить курсор на нужный набор сигналов и с помощью правой кнопки мыши в контекстном меню выбрать пункт **Удалить шаблон** либо нажать клавишу **Del** или кнопку  на панели инструментов;
2. в появившемся окне для подтверждения удаления нажать кнопку **Да**, для отказа – кнопку **Нет**.

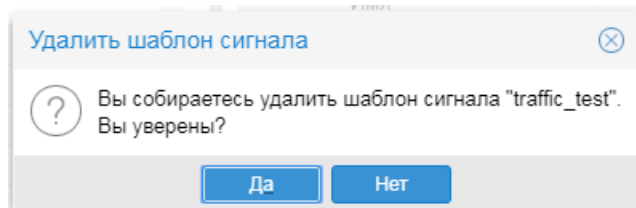


Рисунок 31. Подтверждение удаления набора сигналов

Если набор сигналов не прикреплен ни к одному целевому объекту, то он будет удален из списка на вкладке **Сигналы**. В противном случае выдаётся сообщение об ошибке.

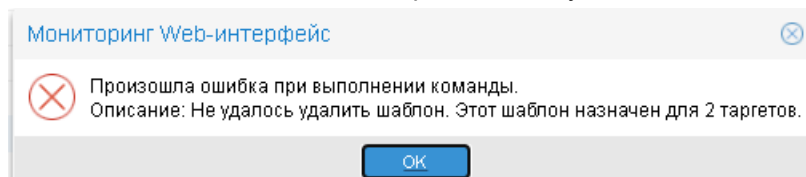


Рисунок 32. Сообщение об ошибке при попытке удалить набор сигналов, прикрепленный к целевым объектам

Для удаления такого набора сигналов следует сначала [открепить](#) его от целевых объектов.

4.2.2. Назначение наборов сигналов целевым объектам

Прикрепление наборов сигналов к целевым объектам и изменение настроек привязки (т.е. изменение локальных значений параметров) пользователь осуществляет через контекстное меню окна **Дерево конфигурации**.

Примечание. В контекстном меню формируются соответствующие пункты, только если для ветви дерева конфигурации в группе пользователей установлено право доступа **Привязка сигналов/оповещений**.

Пользователь может выполнять следующие действия:

- ❖ [Привязка набора сигналов к целевым объектам](#)
- ❖ [Открепление набора сигналов от целевого объекта](#)
- ❖ [Изменение локальных значений параметров сигнала](#)

4.2.2.1. Привязка набора сигналов к целевым объектам

К одному целевому объекту можно привязать только один набор сигналов. После того как набор сигналов привязан к целевому объекту, при необходимости можно установить специфические настройки применения набора сигналов, для этого следует [задать локальные значения параметров сигналов](#).

Привязывать наборы сигналов к целевым объектам можно двумя методами.

Привязка набора сигналов к конкретному целевому объекту

Для выполнения этой операции следует:

1. в окне **Дерево конфигурации** поместить курсор на целевой объект, к которому нужно привязать набор сигналов;
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Привязать сигнал**;

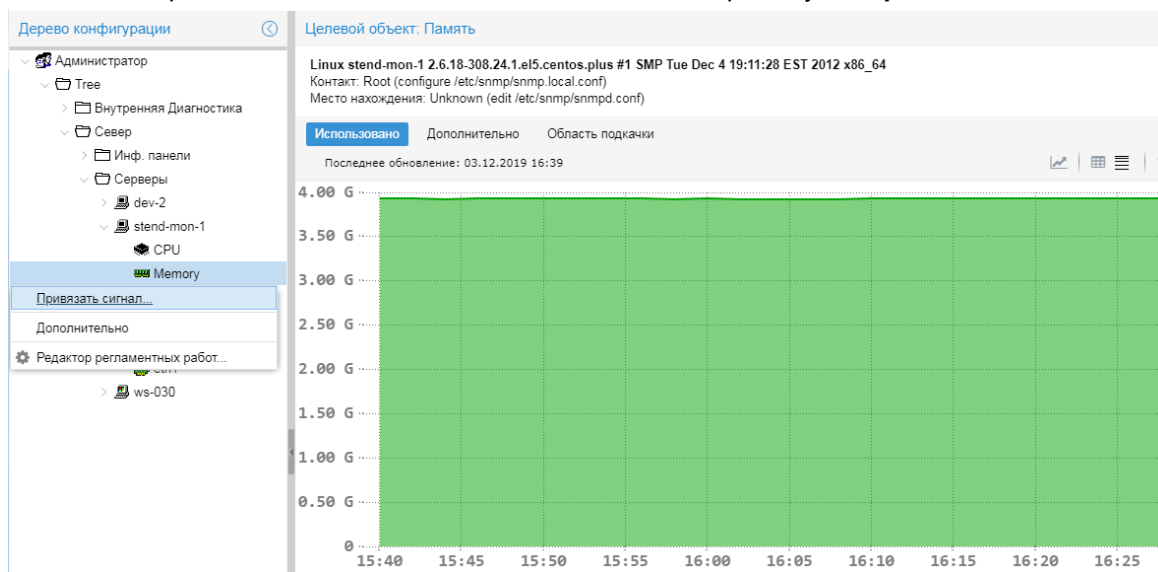


Рисунок 33. Вызов окна для привязки набора сигналов

в результате открывается окно **Привязать сигнал** со списком наборов сигналов, у которых тип назначения совпадает с типом целевого объекта;

3. пометить один набор сигналов;

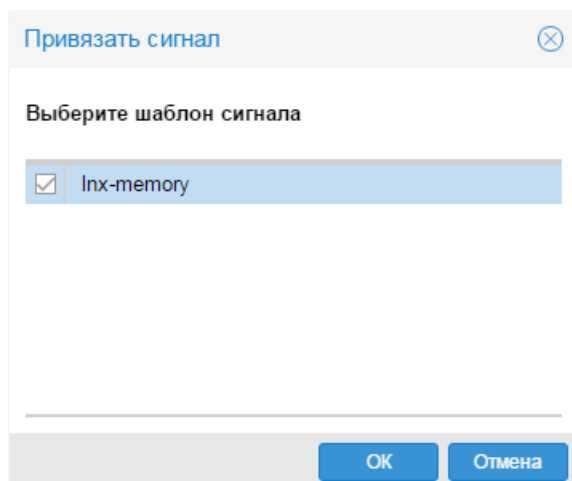


Рисунок 34. Выбор набора сигналов

4. нажать кнопку **ОК**.

В результате в окне **Дерево конфигурации** привязанный набор сигналов отображается как дочерний узел целевого объекта, а на пиктограмме целевого объекта, к которому был привязан набор сигналов, появляется желтый значок.

Дерево конфигурации

- Администратор
 - Tree
 - Внутренняя Диагностика
 - Север
 - Инф. панели
 - Серверы
 - dev-2
 - stend-mon-1
 - CPU
 - Memory
 - Inx-memory**
 - swap_avail
 - used
 - /boot
 - /
 - eth0
 - eth1
 - ws-030

Шаблон сигнала: Inx-memory

Используется для объекта типа: linux-memory-info
Alert for Linux station memory

Состояние сигналов

| Имя | * Источник... | Ст... | Значение | Пороговые значения | Время последней проверки |
|--------------|---------------|-------|----------|--|--------------------------|
| swap_avail | linuxSwap... | ● | NaN | ●:0.00 bytes ●:2.45 Gbytes ●:3.43 Gbytes ●:4.91 Gbytes | 0 |
| использов... | linuxUsed... | ● | NaN | ●:0.00 bytes ●:5.46 Gbytes ●:7.02 Gbytes ●:7.80 Gbytes | 0 |

Локальные параметры сигналов

| Имя | Класс | * Источник данных | Интервал | Функция | Пороговые значения | Описание |
|------------|-------|-------------------|----------|------------|--------------------|----------|
| swap_avail | Gauge | linuxSwapUsed | 600 сек. | максима... | ●:0 ●:50% ●:70% ●: | |
| использ... | Gauge | linuxUsedMemory | 600 сек. | максима... | ●:0 ●:70% ●:90% ●: | |

Глобальные параметры сигналов

| Имя | Класс | * Источник данных | Интервал | Функция | Пороговые значения | Описание |
|------------|-------|-------------------|----------|------------|--------------------|----------|
| swap_avail | Gauge | | 600 сек. | максима... | ●:0 ●:50% ●:70% ●: | |
| использ... | Gauge | | 600 сек. | максима... | ●:0 ●:70% ●:90% ●: | |

Рисунок 35. Привязанный набор сигналов

Привязка наборов сигналов к целевым объектам в рамках выбранного узла дерева конфигурации

Для выполнения этой операции следует:

1. в окне **Дерево конфигурации** поместить курсор на устройство или папку;
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Привязать сигнал**; в результате открывается окно **Привязать сигнал к узлу дерева** со списком существующих наборов сигналов;
3. отметить один или несколько нужных наборов. С помощью кнопки **Выбрать всё** можно отметить сразу все записи, чтобы привязать наборы сигналов, подходящие по типу, ко всем целевым объектам в рамках указанного узла дерева конфигурации:

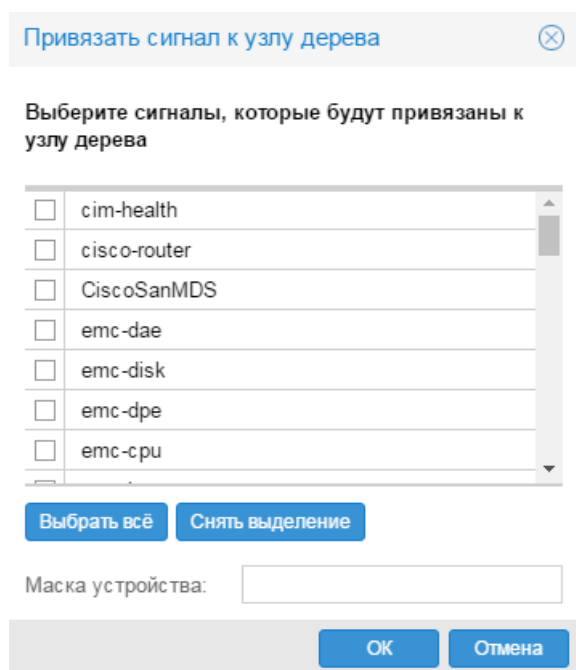


Рисунок 36. Выбор привязываемых наборов сигналов

При необходимости можно настроить привязку набора сигналов по маске имени объекта. Для этого в поле **Маска устройства** следует задать требуемый фильтр в виде стандартного регулярного выражения Perl. Например, маска `cpu.*` позволит привязать сигналы к объектам `cpu`, `cpu1`, `cpu2`, `cpu_total` и т.д. Маска `cpu[0-9]+` позволит привязать сигналы к нумерованным процессорам, то есть `cpu1`, `cpu2` .. `cpu99`;

4. нажать кнопку **ОК**.

В результате выбранные наборы сигналов, подходящие по типу, будут привязаны к целевым объектам в рамках выбранного объекта дерева конфигурации. В окне **Дерево конфигурации** на иконках целевых объектов, к которым привязаны наборы сигналов, появляются желтые значки.

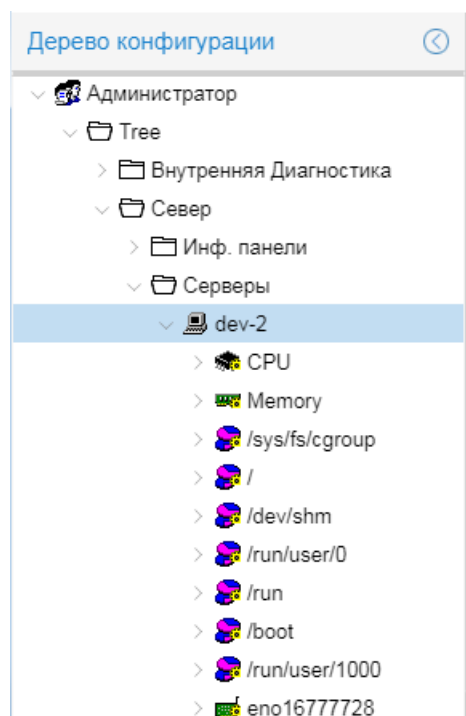


Рисунок 37. Привязанные наборы сигналов

При выборе целевого объекта в окне **Дерево конфигурации** информация о привязанном наборе сигналов отображается в области вывода данных справа.

4.2.2.2. Открепление набора сигналов от целевого объекта

Для открепления наборов сигналов от целевых объектов можно использовать два метода.

Открепление набора сигналов от заданного целевого объекта

Для выполнения этой операции следует:

1. в окне **Дерево конфигурации** поместить курсор на целевой объект, от которого нужно открепить набор сигналов;
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Освободить сигнал**;

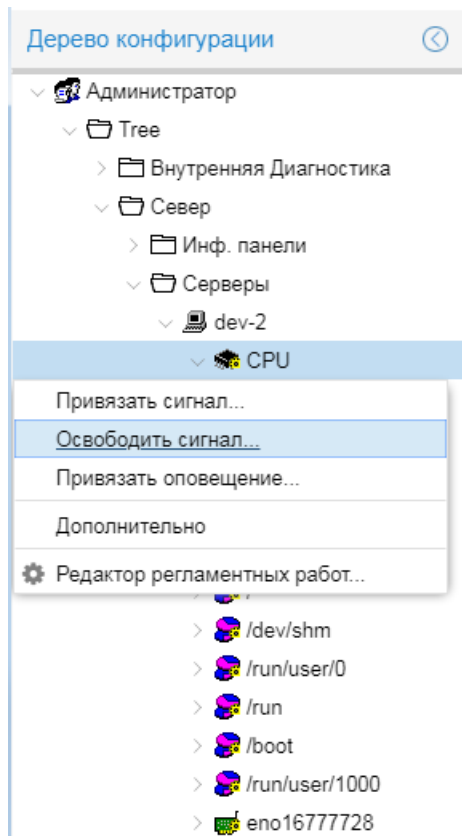


Рисунок 38. Открепление набора сигналов от целевого объекта

3. убедиться, что в появившемся окне нужный набор сигналов отмечен, и нажать кнопку **ОК**.

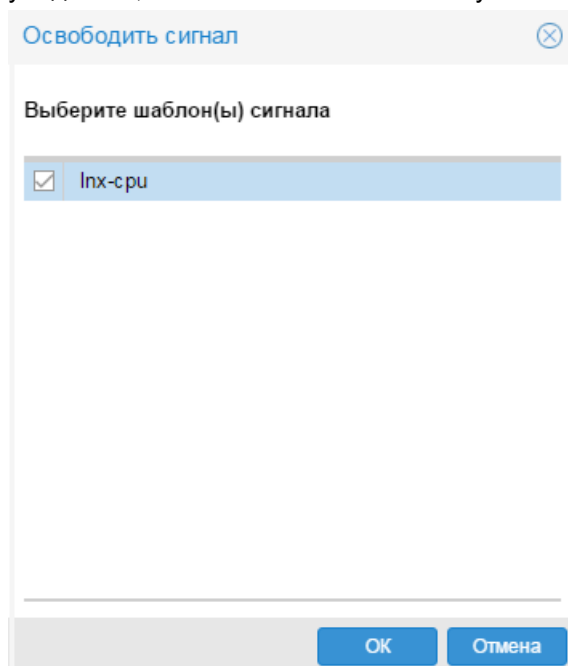


Рисунок 39. Выбор набора сигналов

В результате в окне **Дерево конфигурации** будет удален узел, соответствующий набору сигналов, а с пиктограммы целевого объекта исчезнет желтый значок.

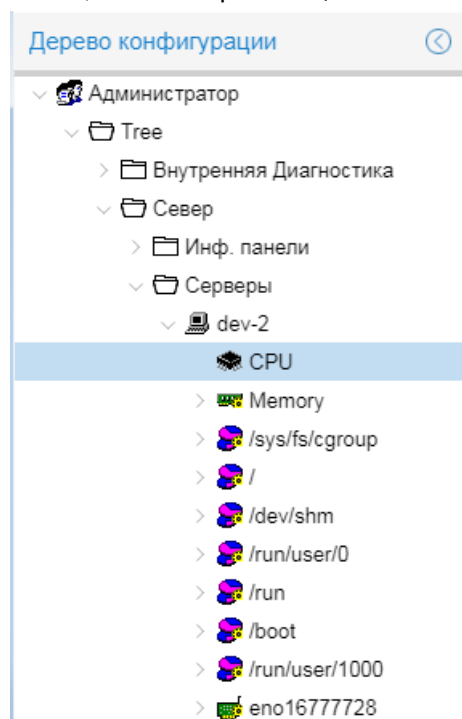


Рисунок 40. Набор сигналов откреплён

Открепление наборов сигналов от целевых объектов в рамках заданного узла дерева конфигурации

Для выполнения этой операции следует:

1. в окне **Дерево конфигурации** поместить курсор на устройство или папку;
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Освободить сигнал**;
3. в появившемся окне пометить один или несколько наборов сигналов, которые необходимо открепить от целевых объектов, либо нажать кнопку **Выбрать все** для открепления наборов сигналов ото всех целевых объектов в рамках выбранного узла дерева конфигурации;

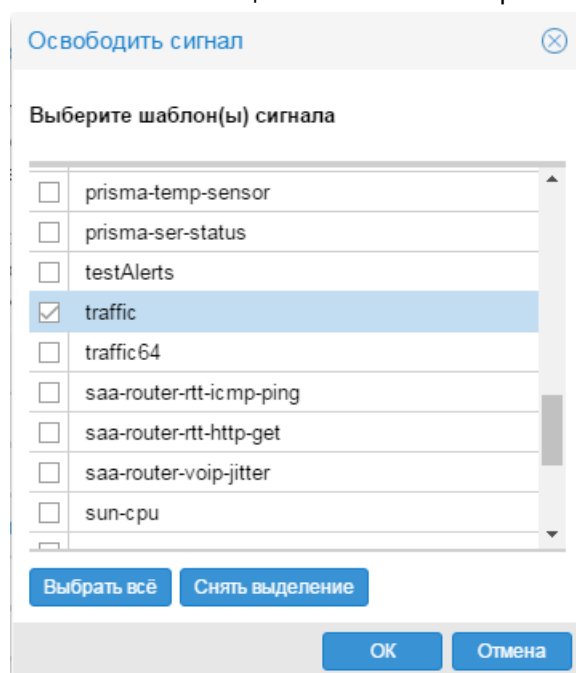


Рисунок 41. Выбор открепляемых наборов сигналов

- нажать кнопку **OK**.

В результате наборы сигналов будут откреплены от всех целевых объектов в рамках выбранного объекта дерева конфигурации. В окне **Дерево конфигурации** с пиктограмм целевых объектов исчезнут желтые значки.

4.2.2.3. Изменение локальных значений параметров сигнала

В наборе сигналов администратор устанавливает [глобальные значения](#) параметров сигнала. После того как набор сигналов привязан к целевому объекту, оператор (пользователь с правами администрирования) имеет возможность задать для параметров сигнала локальные значения, которые будут применяться только для данного целевого объекта.

Для изменения локальных значений параметров сигнала следует:

- в окне **Дерево конфигурации** поместить курсор на сигнал, входящий в набор, привязанный к целевому объекту;
- с помощью правой кнопки мыши в контекстном меню выбрать пункт **Редактировать** (или нажать клавишу **Enter** или дважды кликнуть мышью на выбранном узле дерева конфигурации).

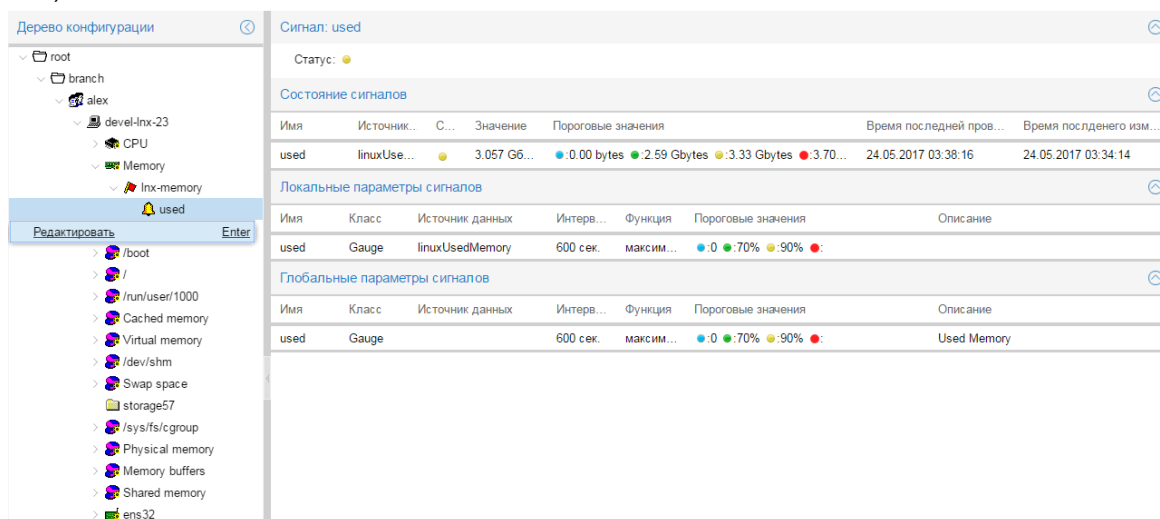


Рисунок 42. Вызов окна для изменения локальных параметров сигнала

В результате в области вывода данных будет показана таблица с параметрами сигнала:

- первый столбец – наименование параметра;
- второй столбец – локальное значение параметра (доступны для изменения);
- третий столбец – глобальное значение параметра, заданное администратором (недоступны для изменения);

Поля формируются в соответствии с типом сигнала. Серые поля недоступны для редактирования;

- изменить локальные значения во втором столбце (по нажатию кнопки **Сбросить** во второй столбец возвращаются последние сохраненные значения);

| Имя | Источн... | С... | Знач... | Пороговые значения | Время последней... | Время послденег... |
|------|-----------|------|-----------|--|---------------------|---------------------|
| used | linuxU... | ● | 3.040 ... | ●:0.00 bytes ●:2.52 Gbytes ●:3.41 Gby... | 24.05.2017 04:14:15 | 24.05.2017 03:34:14 |

Рисунок 43. Просмотр локальных значений параметров сигнала

- по окончании ввода значений для сохранения внесенных изменений нажать кнопку **Сохранить**. В результате в области вывода данных будет показана обновленная информация о сигнале.

| Имя | Класс | Источн... | Инте... | Функ... | Пороговые значения | Описание |
|------|-------|------------------|---------|---------|--------------------|-------------|
| used | Gauge | linuxUsedMemo... | 600 ... | макс... | ●:0 ●:68% ●:92% ●: | Used Memory |

Рисунок 44. Локальные значения параметров сигнала изменены

4.3. Настройка оповещений

Процесс настройки оповещений состоит из двух этапов:

- ❖ настройка наборов оповещений в **Редакторе сигналов и оповещений** выполняется пользователями, входящими в группу с установленной глобальной политикой **Настройка сигналов и оповещений**;

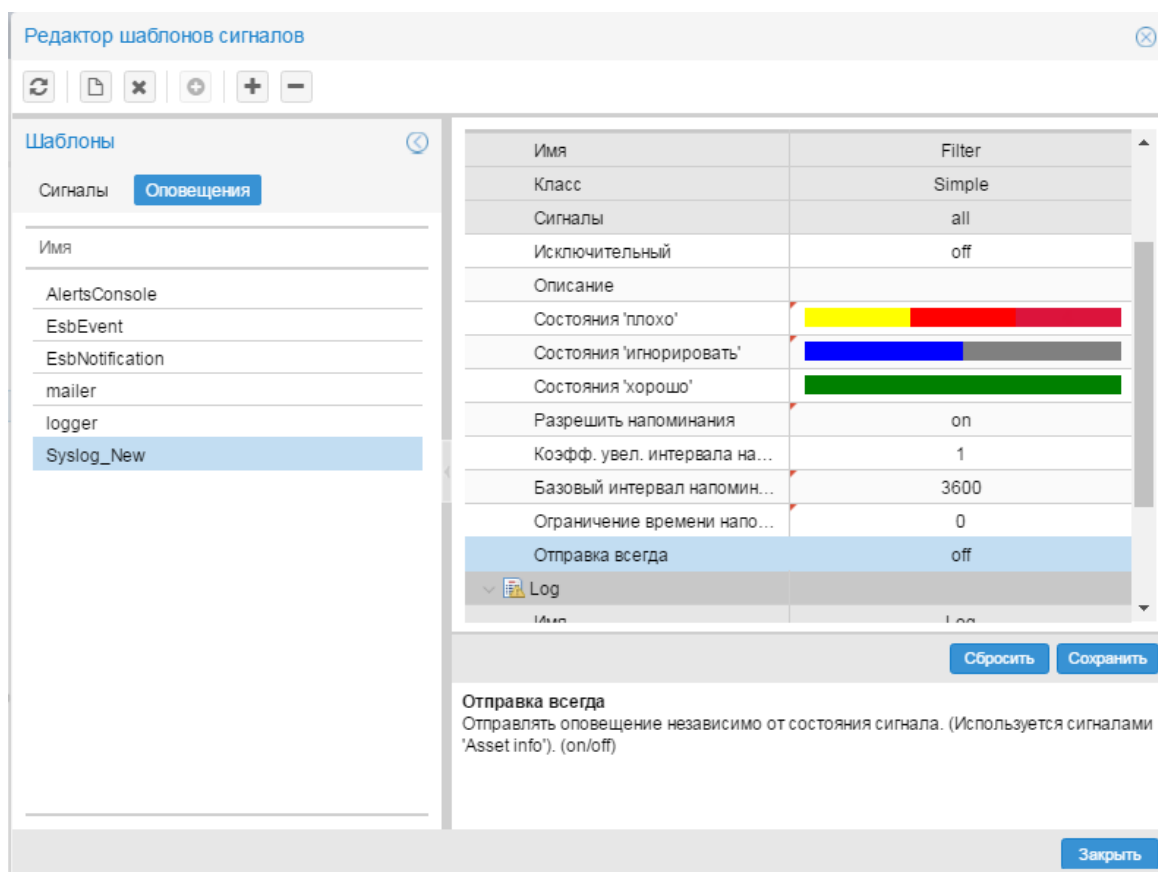


Рисунок 45. Редактор наборов оповещений

- ❖ [назначение наборов оповещений целевым объектам](#) – выполняется через контекстное меню окна **Дерево конфигурации**.

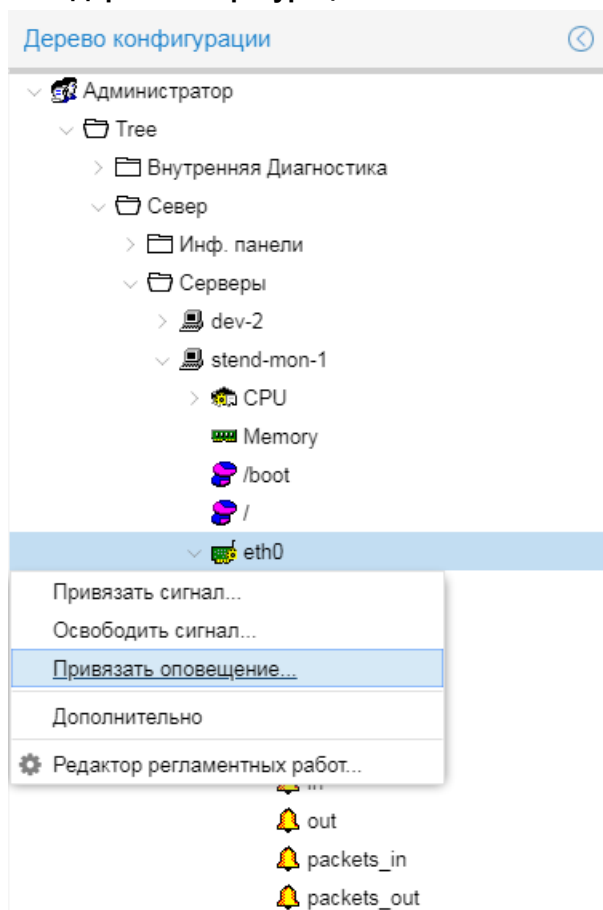



Рисунок 46. Вызов окна для привязки наборов оповещений к целевому объекту

При настройке сигналов следует руководствоваться правилами:

- ❖ набор оповещений обязательно должен содержать один класс оповещений;
- ❖ набор оповещений может содержать один или несколько методов оповещений;
- ❖ к целевому объекту можно прикрепить несколько наборов оповещений.

4.3.1. Настройка наборов оповещений

Настройка наборов оповещений осуществляется в окне **Редактор шаблонов сигналов**, которое вызывается из меню приложений (кнопка  в заголовке Web-интерфейса).

Для настройки наборов оповещений следует перейти на вкладку **Оповещения**.

Чтобы настроить новый набор оповещений, администратору следует:

- ❖ сначала [создать набор оповещений и добавить в него класс оповещений](#);
- ❖ затем [добавить в него методы оповещений](#).


Для ускорения настройки можно [скопировать](#) существующий набор оповещений и внести в него необходимые изменения.

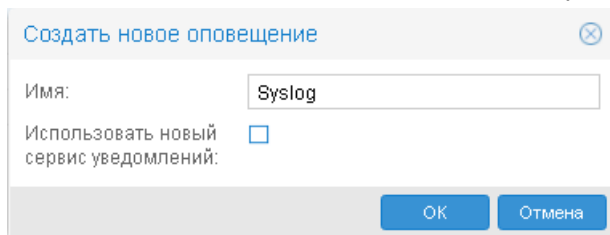
Администратор также имеет следующие возможности:

- ❖ [удалить метод/класс оповещений из набора оповещений](#);
- ❖ [переименовать набор оповещений](#);
- ❖ [изменить глобальные значения параметров метода / класса оповещений](#);
- ❖ [удалить набор оповещений](#).

4.3.1.1. Создание набора оповещений и добавление класса оповещений

Для создания нового набора оповещения следует:

1. в окне **Шаблоны** открыть вкладку **Оповещения** и в контекстном меню выбрать пункт **Новый шаблон**, либо нажать клавишу **Ins** или кнопку  на панели инструментов;
2. в появившемся окне ввести имя нового набора и нажать **ОК**.

**Рисунок 47. Ввод имени нового набора оповещений**

В результате новый пустой набор появится в списке на вкладке **Оповещения**, а рабочая область будет содержать только головной блок параметров;

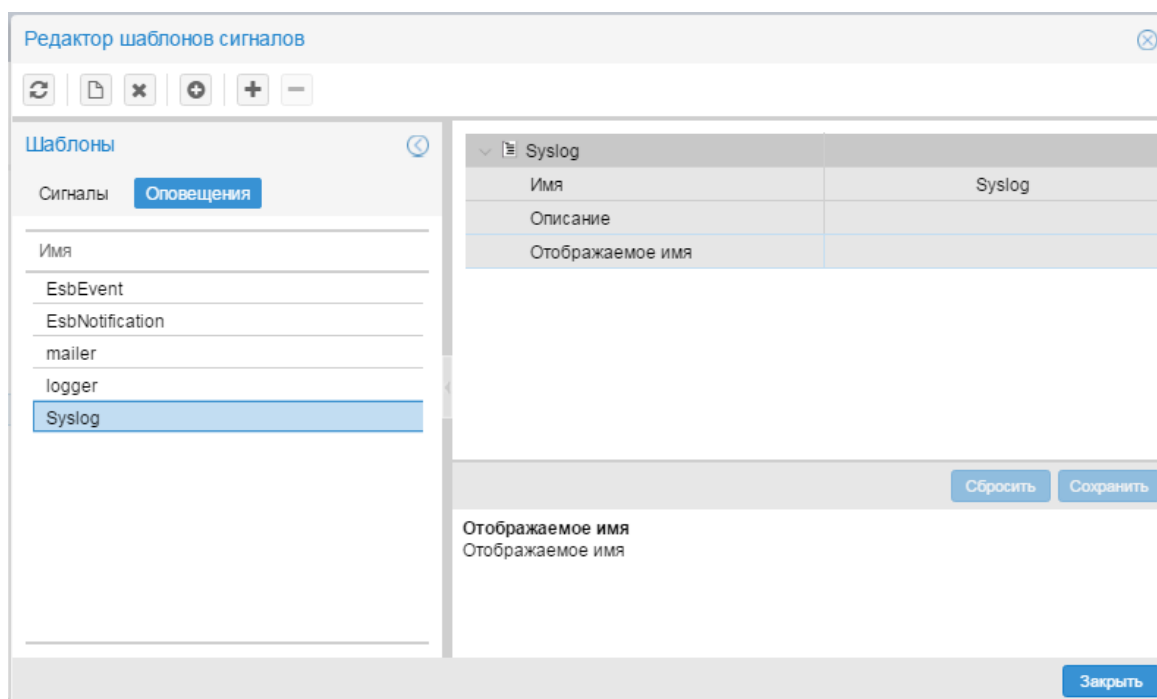



Рисунок 48. Новый набор оповещений в списке

3. чтобы добавить класс оповещений, на вкладке **Оповещения** поместить курсор на только что созданный пустой набор оповещений и выполнить одно из действий:
 - ▶ с помощью правой кнопки мыши в контекстном меню выбрать пункт **Добавить класс** → **Simple**;
 - ▶ на панели инструментов нажать кнопку , в появившемся окне в списке выбрать класс **Simple** и нажать кнопку **ОК**;

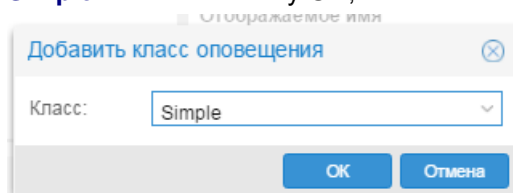


Рисунок 49. Выбор класса оповещений

4. в появившемся окне ввести название класса оповещений и нажать **ОК**;

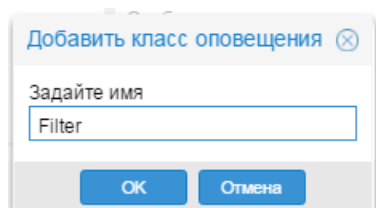


Рисунок 50. Ввод имени класса оповещений

в результате в рабочей области сформируется вложенный блок с параметрами класса оповещений;

5. задать глобальные значения параметров класса оповещений либо оставить значения по умолчанию без изменений; подробное описание параметров см. в справочнике Параметры класса оповещений Simple;
6. нажать кнопку **Сохранить**.

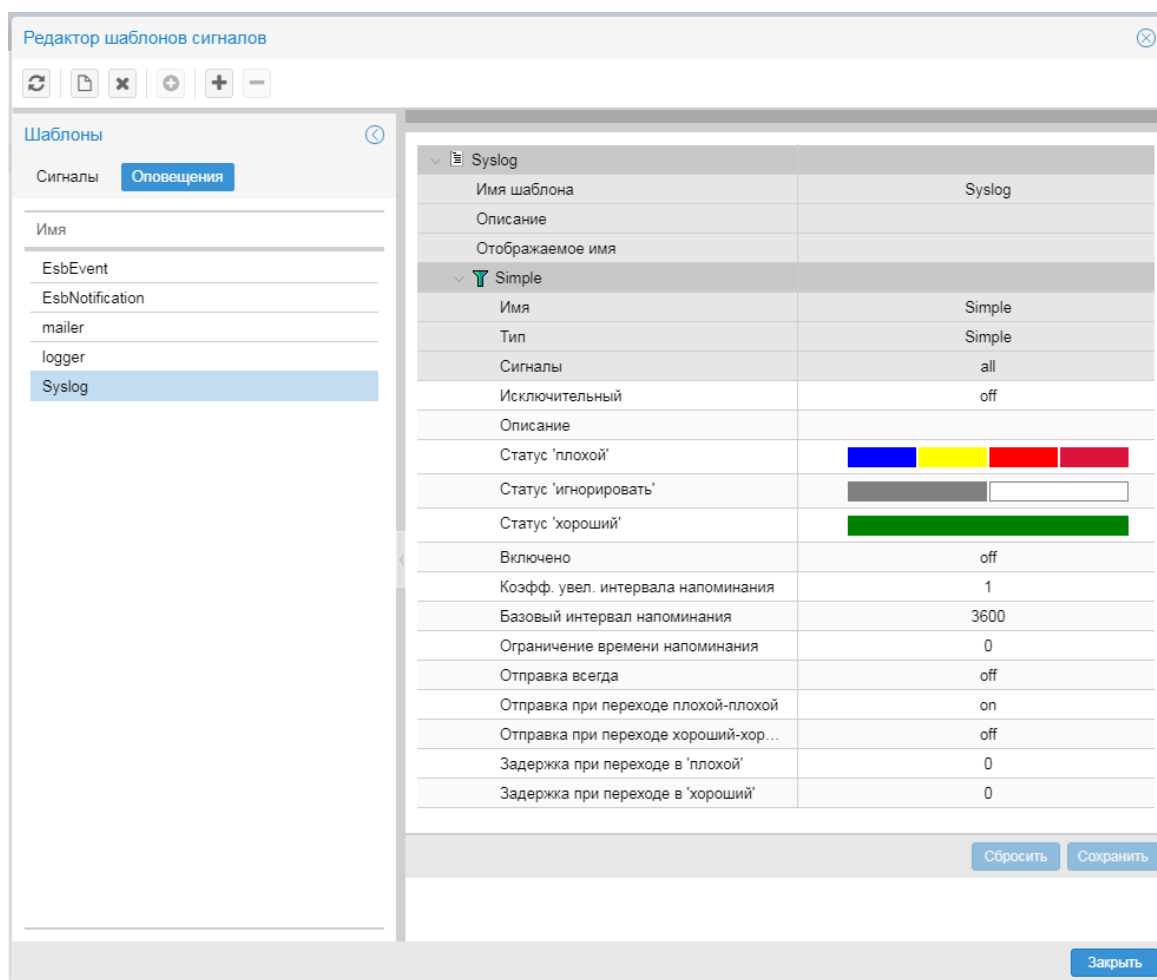



Рисунок 51. Новый набор оповещений с классом

Затем необходимо в набор оповещений [добавить методы оповещения](#).

4.3.1.2. Добавление метода оповещений

Для добавления метода к набору оповещений следует:

1. на вкладке **Оповещения** поместить курсор на набор оповещений, к которому нужно добавить метод оповещений;
2. выполнить одно из действий:
 - ❖ в контекстном меню выбрать пункт **Оповещения / Добавить метод** и далее – подпункт, соответствующий типу оповещений;
 - ❖ на панели инструментов нажать кнопку , в появившемся окне в списке **Метод** выбрать тип оповещений и нажать кнопку **ОК**;

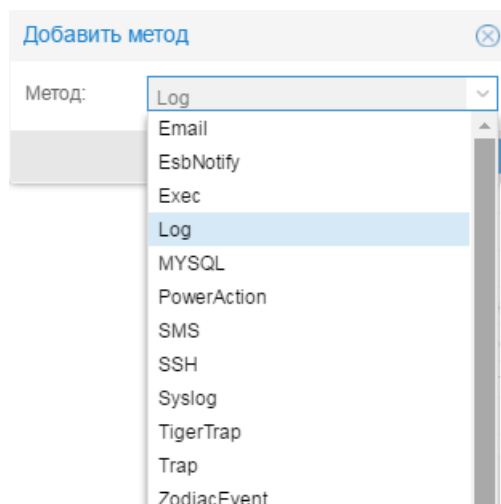


Рисунок 52. Выбор типа оповещений

3. в появившемся окне ввести имя метода и нажать кнопку **ОК**;

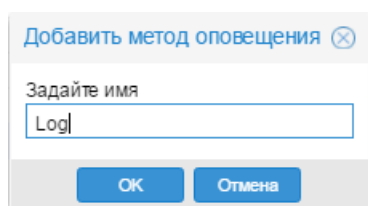


Рисунок 53. Ввод имени метода оповещений

в результате в рабочей области сформируется вложенный блок с параметрами метода оповещений;

4. ввести глобальные значения параметров метода оповещений. Набор полей зависит от выбранного типа оповещений.

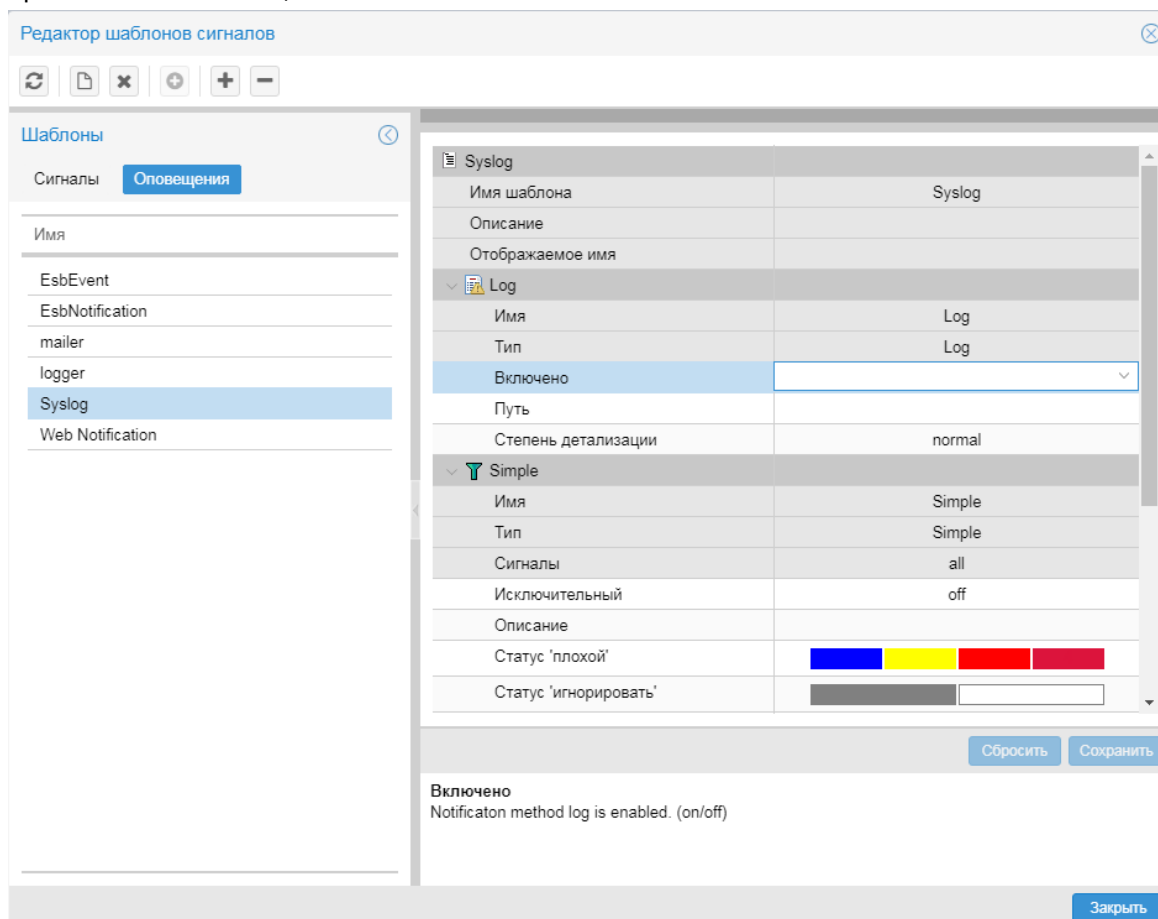



Рисунок 54. Параметры метода оповещений

5. после ввода параметров метода нажать кнопку **Сохранить**.

4.3.1.3. Удаление метода/класса оповещений из набора оповещений

Для удаления метода/класса из набора оповещений следует:

1. на вкладке **Оповещения** поместить курсор на набор оповещений, из которого требуется удалить метод или класс;
2. выполнить одно из действий:
 - ❖ с помощью правой кнопки мыши в контекстном меню выбрать пункт **Удалить метод / Удалить класс** и далее – подпункт с названием метода / класса оповещений;
 - ❖ на панели инструментов нажать кнопку  и в появившемся окне в списке выбрать метод / класс;
3. в появившемся окне для подтверждения удаления нажать кнопку **Да**, для отказа – кнопку **Нет**.

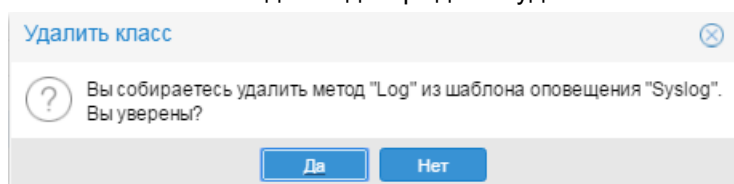


Рисунок 55. Подтверждение удаления

В результате блок с параметрами метода / класса оповещений будет удален из рабочей области.

4.3.1.4. Копирование набора оповещений

Для ускорения и упрощения настройки нового набора оповещений можно создать копию существующего набора оповещений и затем внести в него изменения.

Чтобы создать копию набора оповещений, следует:

1. на вкладке **Оповещения** поместить курсор на набор сигналов, который нужно скопировать;
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Копировать шаблон**;
3. в появившемся окне ввести имя нового набора сигналов; по умолчанию предлагается **<имя исходного набора сигналов>_copu**;

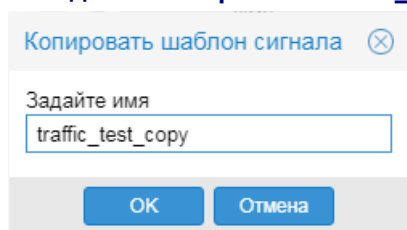


Рисунок 56. Ввод имени нового набора сигналов

4. нажать кнопку **ОК**.

В результате будет создан новый набор оповещений, идентичный исходному набору оповещений.

Для дальнейшей настройки набора сигналов можно выполнить следующие действия:

- ❖ [Переименование набора оповещений](#)
- ❖ [Добавление метода оповещений](#)
- ❖ [Изменение глобальных значений параметров метода / класса оповещений](#)

4.3.1.5. Переименование набора оповещений

Переименование набора оповещений – это изменение его отображаемого имени.

Чтобы переименовать набор оповещений, следует:

1. на вкладке **Оповещения** поместить курсор на нужный набор сигналов;

2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Переименовать шаблон** или нажать клавишу **F2**;
3. в появившемся окне ввести новое название набора сигналов;
4. нажать кнопку **ОК**.

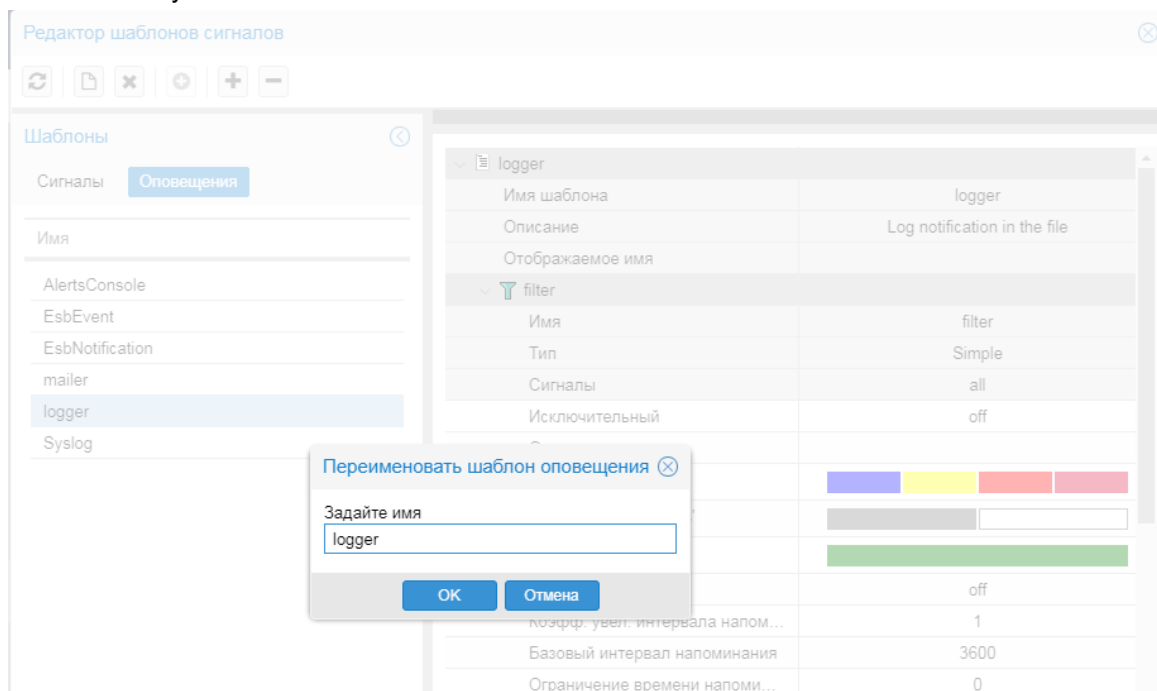


Рисунок 57. Переименование набора оповещений

4.3.1.6. Изменение глобальных значений параметров метода / класса оповещений

Измененные глобальные значения параметров метода/класса оповещений распространяется на целевые объекты, к которым привязан соответствующий набор оповещений без указания локальных значений.

Для изменения глобальных значений параметров метода / класса оповещений следует:

1. на вкладке **Оповещения** поместить курсор интересующий набор оповещений;
2. в рабочей области раскрыть блок с параметрами нужного метода / класса оповещений и изменить значения полей во втором столбце. Поля в блоке метода оповещений формируются в соответствии с типом оповещений. Поля на затененном фоне недоступны для редактирования. Подробное описание полей см. в справочниках: Параметры класса оповещений Simple и Типы оповещений;
3. для сохранения изменений нажать кнопку **Сохранить**, для отказа от изменений – кнопку **Сбросить**.

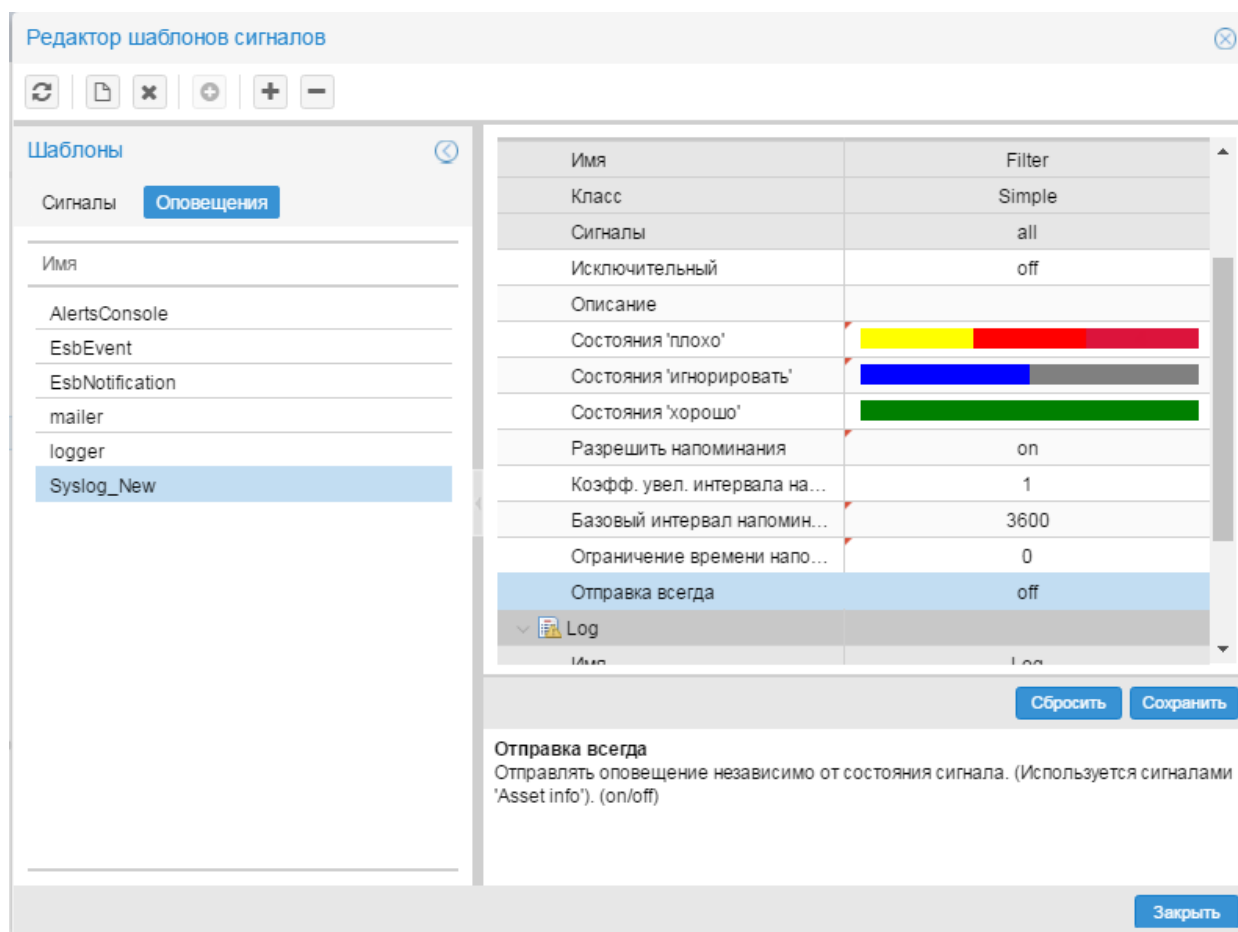



Рисунок 58. Изменение глобальных значений параметров оповещений

4.3.1.7. Удаление набора оповещений

Допускается удаление наборов оповещений, не прикрепленных к целевым объектам.

Чтобы удалить набор оповещений, следует:

1. на вкладке **Оповещения** поместить курсор на нужный набор сигналов и с помощью правой кнопки мыши в контекстном меню выбрать пункт **Удалить шаблон** либо нажать кнопку  на панели инструментов или клавишу **Del**;
2. в появившемся окне для подтверждения удаления нажать кнопку **Да**, для отказа – кнопку **Нет**.

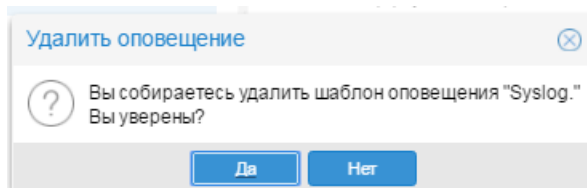


Рисунок 59. Подтверждение удаления набора сигналов

Если набор оповещений не прикреплен ни к одному целевому объекту, то он будет удален из списка на вкладке **Оповещения**. В противном случае выдается сообщение об ошибке.

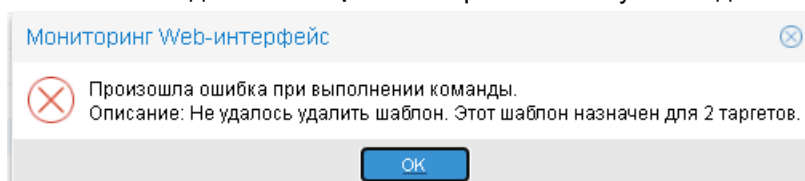


Рисунок 60. Сообщение об ошибке при попытке удалить набор оповещений, прикрепленный к целевым объектам

Для возможности удаления такого набора оповещений следует сначала [открепить](#) его от целевых объектов.

4.3.2. Назначение наборов оповещений целевым объектам

Прикрепление наборов оповещений к целевым объектам и изменение настроек привязки (т.е. изменение локальных значений параметров) пользователь осуществляет через контекстное меню окна **Дерево конфигурации**.

Примечание. В контекстном меню формируются соответствующие пункты, только если для ветви дерева конфигурации в группе пользователей установлено право доступа **Привязка сигналов/оповещений**.

Администратор может выполнять следующие действия:

- ❖ [Привязка набора оповещений к целевым объектам](#)
- ❖ [Открепление набора оповещений от целевого объекта](#)
- ❖ [Изменение локальных значений параметров оповещений](#)
- ❖ [Проверка формирования оповещений](#)

4.3.2.1. Привязка набора оповещений к целевым объектам

К одному целевому объекту можно привязать несколько наборов оповещений. После того как набор оповещений привязан к целевому объекту, при необходимости можно установить специфические настройки применения набора оповещений, для этого следует задать локальные значения параметров оповещений.

Осуществлять привязку наборов оповещений можно как к конкретному целевому объекту, так и сразу ко всем целевым объектам в рамках выбранного устройства, папки или учетной записи.

Для выполнения этой операции следует:

1. в окне **Дерево конфигурации** поместить курсор на узел дерева, для которого нужно осуществить привязку наборов оповещений; таким узлом может быть учетная запись, папка, устройство, целевой объект.
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Привязать оповещение**;

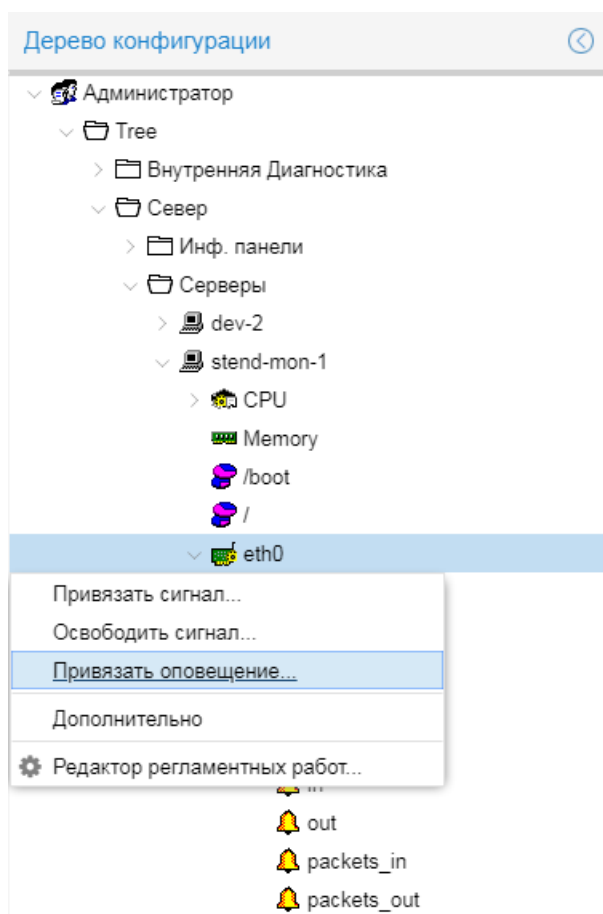


Рисунок 61. Вызов окна для привязки наборов оповещений к целевому объекту

в результате открывается окно **Привязать оповещение** со списком наборов оповещений;

3. отметить флагом один или несколько наборов оповещений для привязки к целевым объектам в рамках узла, выбранного на шаге 1;

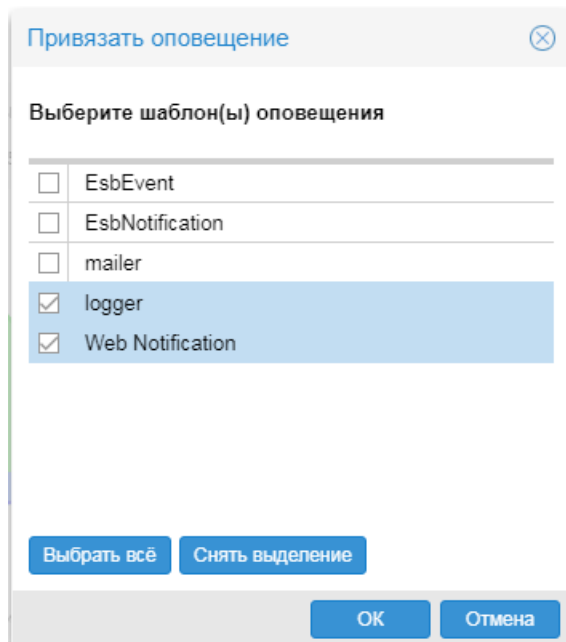


Рисунок 62. Выбор наборов оповещений для привязки к целевому объекту

4. нажать кнопку **ОК**.

В результате в окне **Дерево конфигурации** привязанные наборы оповещений будут показаны как дочерние узлы целевых объектов. На пиктограммах целевых объектов, к которым были привязаны наборы оповещений, появится белый значок.

Дерево конфигурации

- Администратор
 - Tree
 - Внутренняя Диагностика
 - Север
 - Инф. панели
 - Серверы
 - dev-2
 - stend-mon-1
 - CPU
 - Memory
 - /boot
 - /
 - eth0
 - traffic64
 - logger
 - Web Notification
 - filter
 - method
 - eth1
 - ws-030
 - Загрузка серверов

Шаблон оповещения: Web Notification

WebNotification

| | | |
|---------------------------------------|---|---|
| Имя шаблона | WebNotification | |
| Описание | | |
| Отображаемое имя | Web Notification | |
| filter | | |
| Имя | filter | |
| Тип | Simple | |
| Сигналы | all | |
| Исключительный | off | |
| Описание | | |
| Статус 'плохой' | | <div><div></div><div></div><div></div><div></div></div> |
| Статус 'игнорировать' | | <div><div></div><div></div></div> |
| Статус 'хороший' | | <div><div></div></div> |
| Включено | on | |
| Коефф. увел. интервала напоминания | 1 | |
| Базовый интервал напоминания | 3600 | |
| Ограничение времени напоминания | 0 | |
| Отправка при любом изменении статуса | off | |
| Отправка при переходе хороший-плохой | on | |
| Отправка при переходе хороший-хороший | off | |
| Отправка при переходе плохой-хороший | on | |
| Отправка при переходе плохой-плохой | off | |
| Задержка при переходе в 'плохой' | 0 | |
| Задержка при переходе в 'хороший' | 0 | |
| method | | |
| Имя | method | |
| Тип | WsNotify | |
| Включено | on | |
| Шаблон | {{ target_display_name }}: {{ gettext(state_str) }} {{ state_b... | |
| Автоматическое скрытие | on | |
| Задержка | 3000 | |

Проверить

Сбросить

Сохранить

Рисунок 63. Привязанные наборы оповещений

4.3.2.2. Открепление набора оповещений от целевого объекта

Осуществлять открепление наборов оповещений можно как от конкретного целевого объекта, так и сразу от всех целевых объектов в рамках выбранного устройства, папки или учетной записи.

Для выполнения этой операции следует:

1. в окне **Дерево конфигурации** поместить курсор на объект дерева, для которого нужно осуществить открепление наборов оповещений; таким узлом может быть учетная запись, папка, устройство, целевой объект;
2. с помощью правой кнопки мыши в контекстном меню выбрать пункт **Освободить оповещение**;

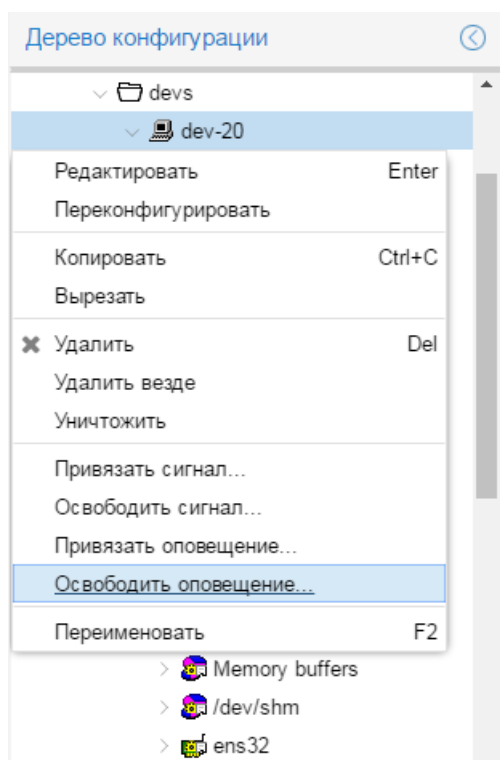


Рисунок 64. Открепление набора оповещений от целевого объекта

3. в появившемся окне отметить один или несколько набор оповещений и нажать кнопку **ОК**.

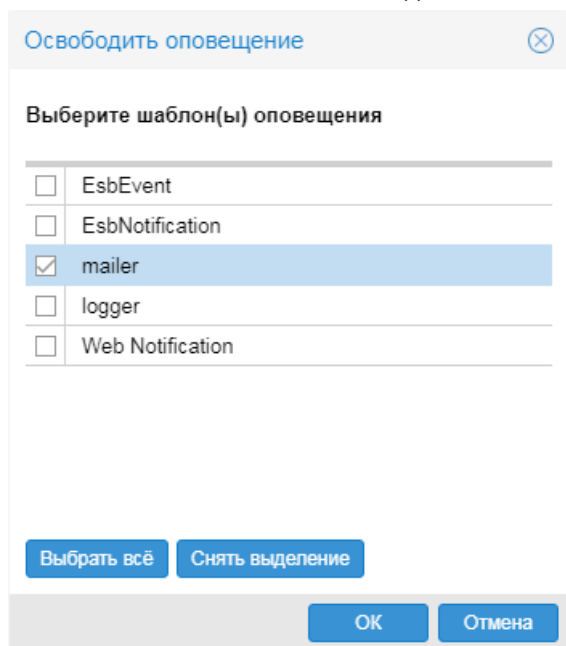


Рисунок 65. Выбор набора оповещений для открепления

В результате отмеченные наборы оповещений будут откреплены от всех целевых объектов в рамках выбранного объекта дерева конфигурации. В окне **Дерево конфигурации** с пиктограмм целевых объектов, от которых откреплены все наборы оповещений, исчезнут белые значки.

4.3.2.3. Изменение локальных значений параметров оповещений

В наборе оповещений администратор устанавливает [глобальные значения](#) параметров класса и метода оповещений. После того как набор оповещений привязан к целевому объекту, пользователь с правами администрирования имеет возможность задать локальные значения, которые будут применяться только для данного целевого объекта.

Для изменения локальных значений параметров класса / метода оповещений следует:

1. в окне **Дерево конфигурации** поместить курсор набор оповещений, привязанный к целевому объекту, либо на входящий в его состав класс или метод оповещений. В зависимости от выбранного узла в области вывода данных будет показана таблица с параметрами набора оповещений и/или входящих в его состав класса и метода (или нескольких методов):
 - ❖ первый столбец – наименование параметра;
 - ❖ второй столбец – локальное значение параметра (доступны для изменения);
 - ❖ третий столбец – глобальное значение параметра, заданное администратором (недоступны для изменения);
2. изменить локальные значения во втором столбце (по нажатию кнопки **Сбросить** во второй столбец возвращаются последние сохраненные значения);

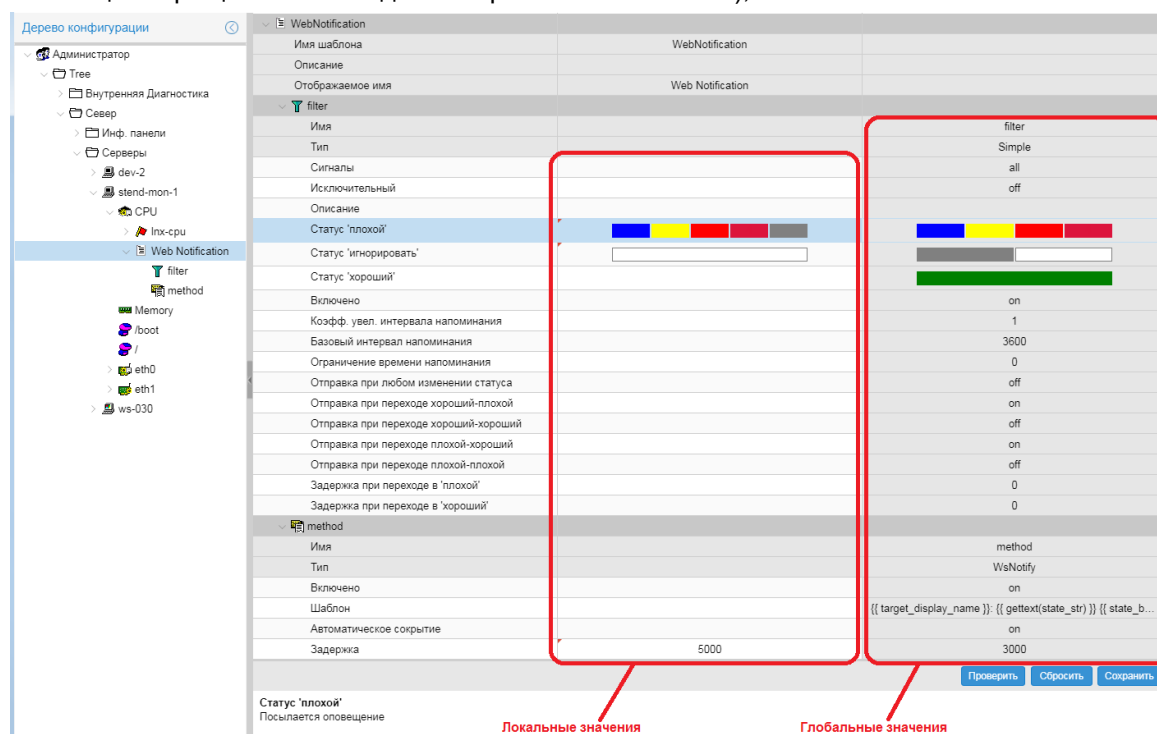


Рисунок 66. Просмотр локальных значений параметров оповещений

3. по окончании ввода значений для сохранения внесенных изменений нажать кнопку **Сохранить**.

4.3.2.4. Проверка формирования оповещений

Чтобы удостовериться в правильности настройки оповещений, администратор может протестировать формирование оповещений по целевому объекту на примере конкретной ситуации (тест-кейса). Для этого следует:

1. в дереве конфигурации поместить курсор на набор оповещений, прикрепленный к целевому объекту;
2. в области вывода данных нажать кнопку **Проверить**;

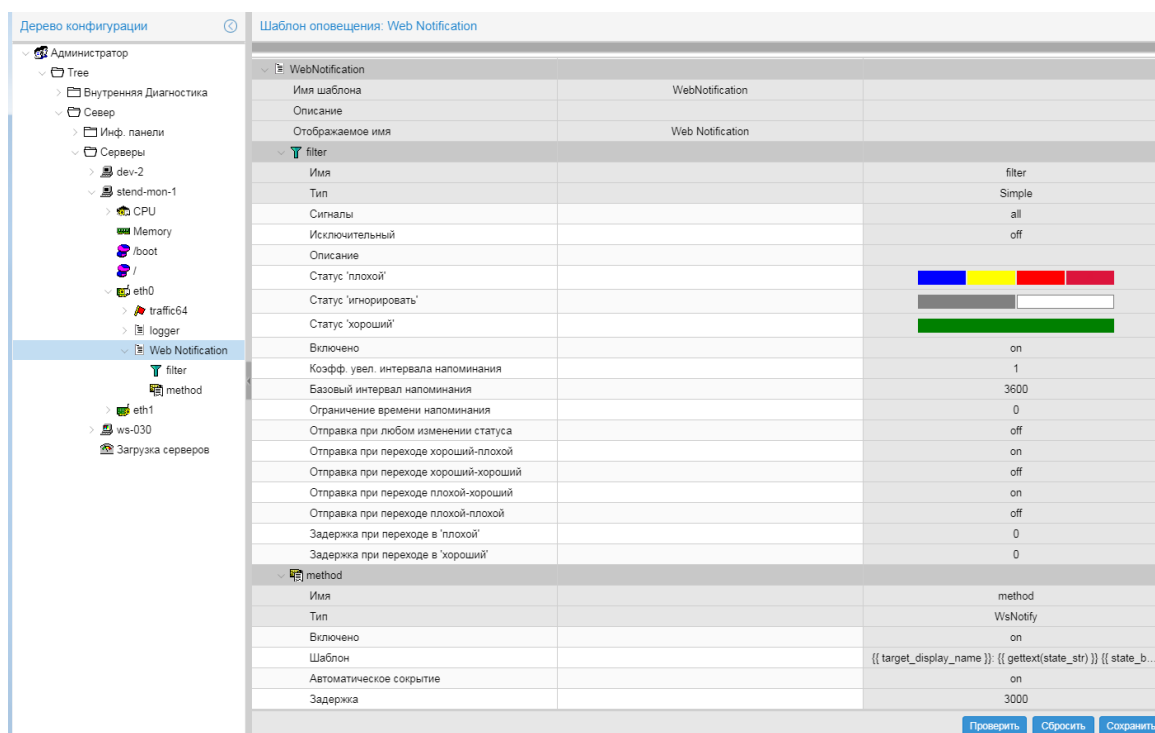


Рисунок 67. Выбор привязанного набора оповещений для проверки

3. в появившемся окне **Проверка уведомлений** ввести параметры, моделирующие поведение сигнала:
 - ❖ **Старое состояние** – начальный статус сигнала;
 - ❖ **Новое состояние** – конечный статус сигнала;
 - ❖ **Выберите сигналы** – сигналы, участвующие в проверке. Предлагается множество сигналов, входящих в состав набора сигналов, привязанного к целевому объекту. Можно удалить часть сигналов или оставить первоначальное множество без изменения;
4. нажать кнопку **Отправить**.

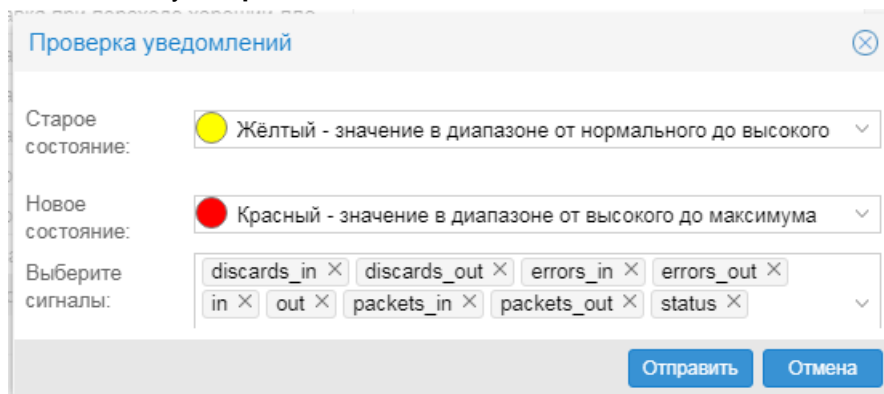


Рисунок 68. Параметры для проверки формирования оповещения (тест-кейс)

В результате по заданному тест-кейсу будут сформированы оповещения (одно или несколько) в соответствии с настройками набора оповещений (например: отправлено письмо на электронную почту, сформирована запись в лог-файле, появится всплывающее уведомление). По итогам тестирования администратор может внести изменения в настройки.

4.3.3. Настройка универсальных шаблонов оповещений

Настройка [универсальных шаблонов оповещений](#) осуществляется в окне **Редактор шаблонов сигналов** на вкладке **Оповещения**. Общий подход аналогичен настройке наборов оповещений. См. далее:

[Создание нового шаблона](#)


[Особенности ввода некоторых параметров шагов](#)

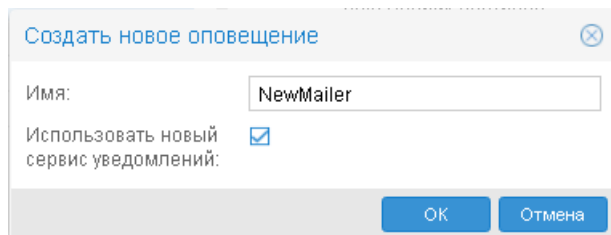
[Пример универсального шаблона](#)

Настроенный шаблон можно привязать к целевым объектам аналогично наборам оповещений. См. [Назначение наборов оповещений целевым объектам](#).

4.3.3.1. Создание нового шаблона

Для создания универсального шаблона оповещения следует:

1. в окне **Шаблоны** открыть вкладку **Оповещения** и в контекстном меню выбрать пункт **Новый шаблон**, либо нажать клавишу **Ins** или кнопку  на панели инструментов;
2. в появившемся окне ввести название нового шаблона, установить флаг **Использовать новый сервис уведомлений** и нажать **ОК**.



3. **Рисунок 69. Ввод названия нового шаблона оповещений**

В результате новый пустой шаблон появится в списке на вкладке **Оповещения**, а рабочая область будет содержать только головной блок параметров.

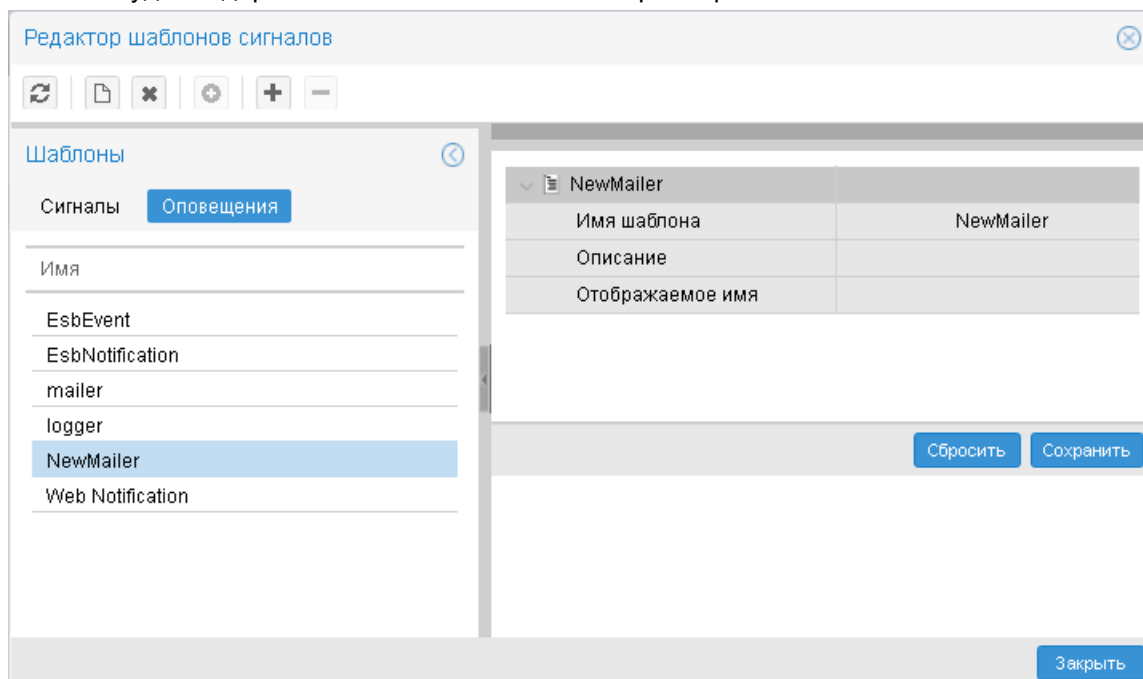



Рисунок 70. Новый шаблон оповещений в списке

3. для добавления шага в шаблон оповещений выполнить одно из действий:
 - ❖ в контекстном меню выбрать пункт **Оповещения** → **Добавить метод** и далее – подпункт, соответствующий типу шага;
 - ❖ на панели инструментов нажать кнопку , в появившемся окне в списке **Шаг** выбрать тип шага и нажать кнопку **ОК**.

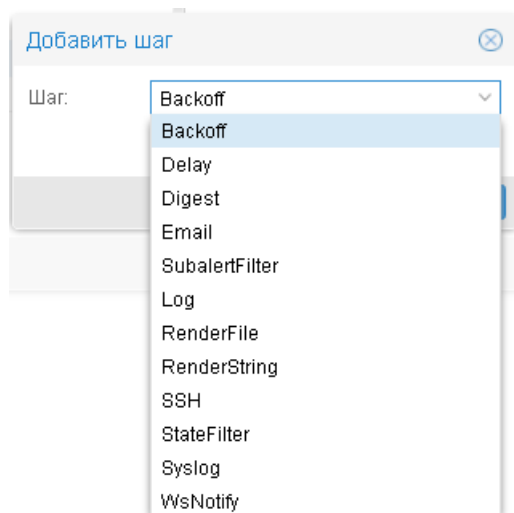


Рисунок 71. Выбор типа шага

4. в появившемся окне ввести имя шага и нажать кнопку **ОК**;

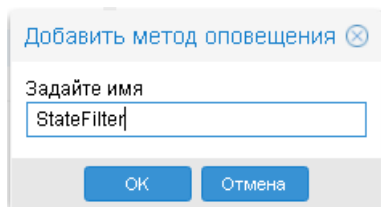


Рисунок 72. Ввод имени шага

в результате в рабочей области сформируется вложенный блок для ввода параметров шага;

5. добавить другие необходимые шаги, повторив пункты 3 и 4;
6. для каждого шага ввести глобальные значения параметров и настроить зависимости. Набор параметров зависит от типа шага. Одинаковые параметры для всех типов шагов:

Описание – комментарий;

Включено – признак активности шага: **on** – шаг используется, **off** – отключен.

Стрелка перед названием параметра означает, что данный параметр является зависимым, т.е. его значение может быть передано из других шагов.

См. также [Особенности ввода некоторых параметров шагов](#).

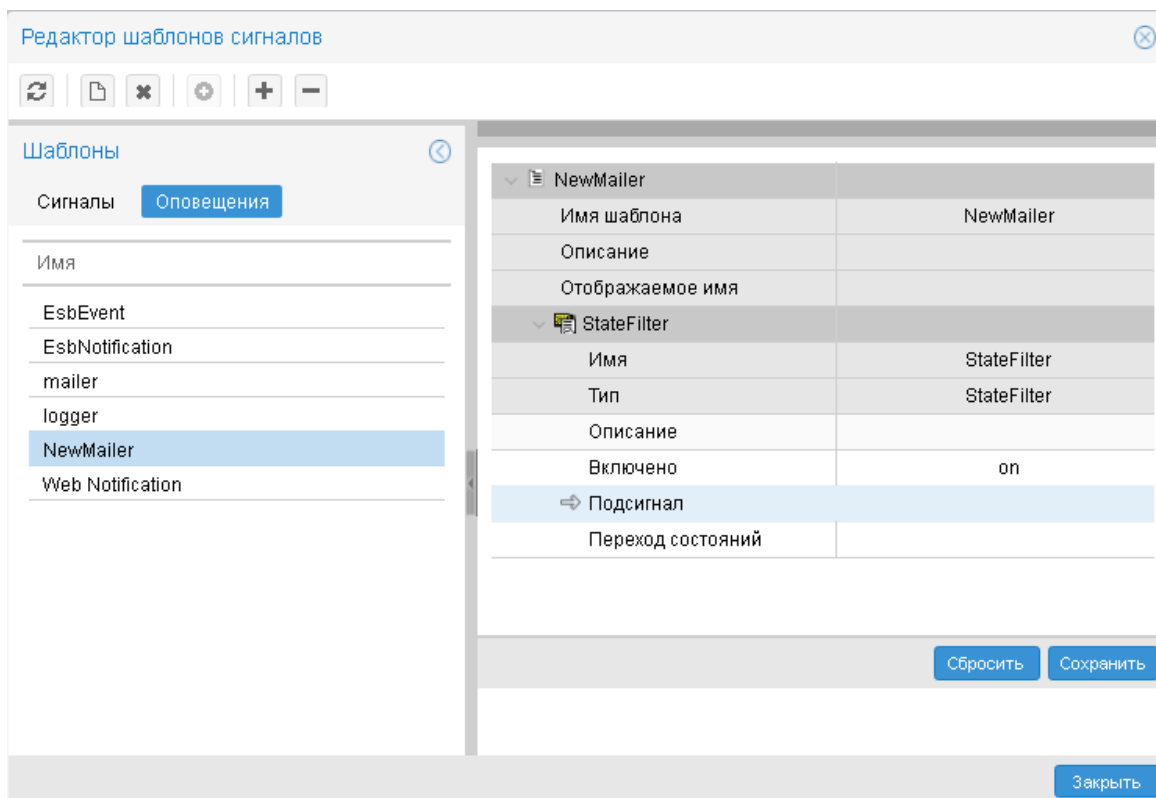


Рисунок 73. Параметры шага

7. после ввода параметров шага нажать кнопку **Сохранить**.

4.3.3.2. Особенности ввода некоторых параметров шагов

При редактировании параметра, который может зависеть от предыдущего шага, появляется окно для настройки зависимостей. Значения поля **Зависит от** имеют формат: `[[имя_шага/выход]]`. Если выход не указан, то по умолчанию используется скрытое `return`.

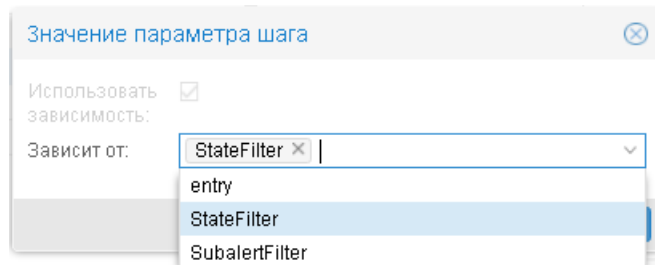


Рисунок 74. Настройка зависимостей

Некоторые поля поддерживают два режима: статический и динамический. В этом случае можно задать значение как зависимость от некоторого шага, так и вручную статическими данными. Режим переключается с помощью флага **Использовать зависимость**. Для полей, поддерживающих только режим зависимости, этот флаг установлен и недоступен для изменения.

Первый шаг цепочки должен иметь зависимость `entry`.

Важно: для генерации шаблонов всегда передается массив объектов типа `Subalert`, поэтому при необходимости использования одиночного сигнала требуется указать:

```
{! set subalert = subalerts[0] !}
```

Пример

1. Входящий сигнал одиночный, тогда шаблон будет выглядеть следующим образом:

```
{! set subalert = subalerts[0] !}name: {{ subalert.name }}; state: {{ sub-alert.state }}
```

- На вход подается массив сигналов, например, из шага **Digest**. В таком случае можно использовать следующий шаблон:

```
{! for subalert in subalerts !}name: {{ subalert.name }}; state: {{ subalert.state }}{! endfor !}
```

4.3.3.3. Пример универсального шаблона

По шаблону, показанному на рисунке ниже, системный процесс `notifng` выполняет следующие действия:

- принимает все сигналы, переходящие из состояния 1 в 3 (шаг **StateFilter**, поле **Переход состояний**);
- выполняет задержку в 5 секунд (шаг **Delay**);
- отправляет уведомление во всплывающем окне с текстом из поля **Содержимое** (шаг **ws**).





| | |
|---|--|
| ▼  Delay | |
| Имя | Delay |
| Тип | Delay |
| Описание | |
| Включено | on |
| ⇒ Подсигнал | [[StateFilter/out1]] |
| Задержка | 5 |
| ▼  StateFilter | |
| Имя | StateFilter |
| Тип | StateFilter |
| Описание | |
| Включено | on |
| ⇒ Подсигнал | [[entry]] |
| Переход состояний |  ('1':{"3":["out1","out2"]} ... |
| ▼  ws | |
| Имя | ws |
| Тип | WsNotify |
| Описание | |
| Включено | on |
| ⇒ Подсигнал | [[Delay]] |
| ⇒ Содержимое | Content |
| Автоматическое сокрытие | on |
| Задержка | 3000 |

Рисунок 75. Шаги в универсальном шаблоне

4.4. Редактор информационных сообщений

Web-приложение **Редактор информационных сообщений** позволяет настраивать шаблоны информационных сообщений и отправлять сообщения пользователям ПП PhoenixDS. Web-приложение доступно администраторам из группы пользователей с установленной глобальной политикой **(i) Управление информационными сообщениями**.

Для вызова **Редактора информационных сообщений** следует в заголовке Web-интерфейса

нажать кнопку  и в меню приложений нажать иконку с названием приложения.

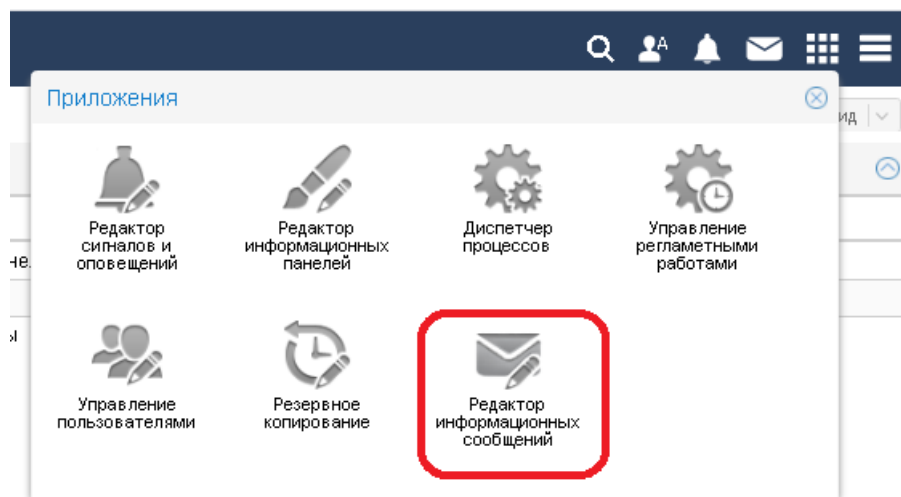


Рисунок 76. Вызов Редактора информационных сообщений

4.4.1. Основное окно

Основное окно приложения **Редактор информационных сообщений** имеет следующую структуру:

- ❖ под заголовком окна размещена панель инструментов;
- ❖ в окне **Шаблоны сообщений** выводится перечень настроенных шаблонов сообщений;
- ❖ справа в рабочей области отображаются параметры шаблона, выбранного в окне **Шаблоны сообщений**.

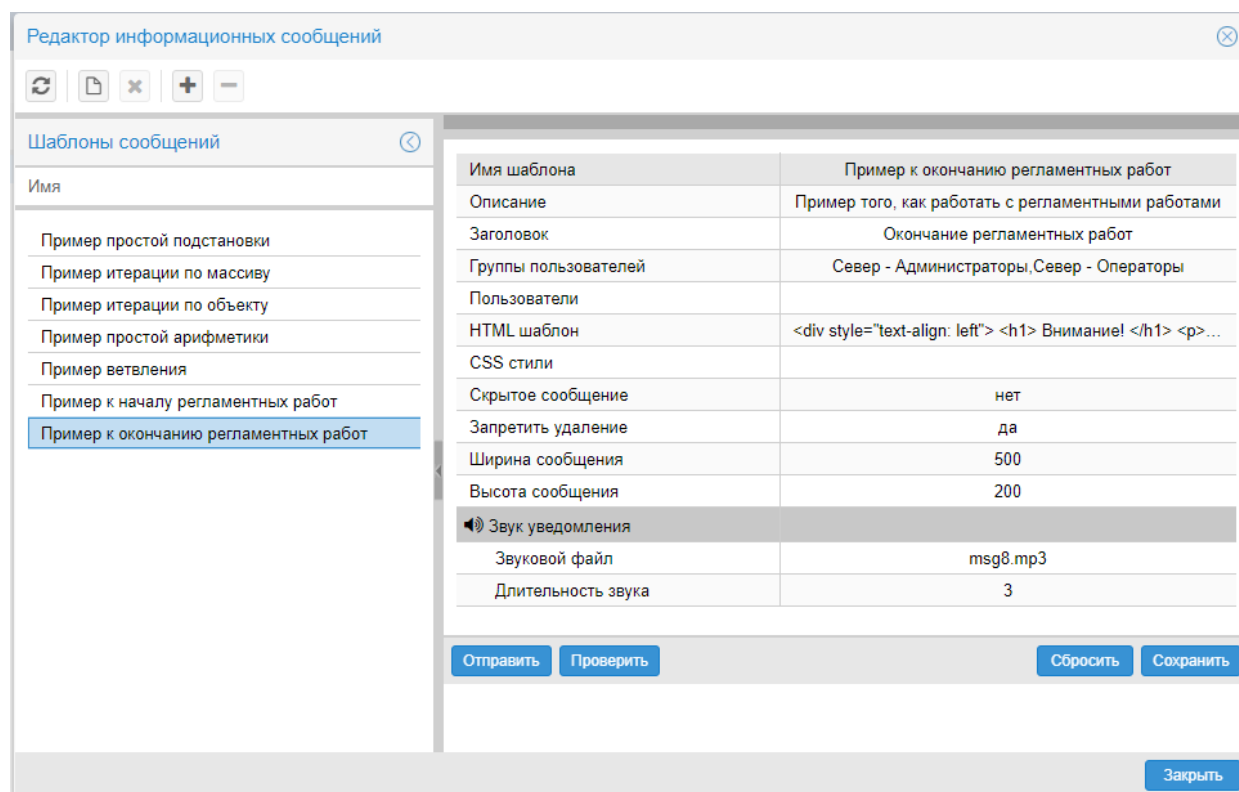



Рисунок 77. Редактор информационных сообщений

Для выхода из окна **Редактор информационных сообщений** можно использовать кнопку  в заголовке окна или кнопку **Заккрыть** в правом нижнем углу окна.

Панель инструментов

В окне **Редактор информационных сообщений** для выполнения действий предусмотрены кнопки на панели инструментов, комбинации клавиш и пункты контекстного меню окна **Шаблоны сообщений**. Некоторые действия можно выполнить несколькими способами. Чтобы вызвать контекстное меню, следует в окне **Шаблоны сообщений** поместить курсор на строку списка и кликнуть правой кнопкой мыши.

Ниже перечислены кнопки панели инструментов и соответствующие им пункты контекстного меню и клавиши.



Обновление информации в окне **Редактор информационных сообщений** в соответствии с данными сервера PhoenixDS.



| **Новый шаблон...** | **Ins**

[Создание шаблона информационных сообщений.](#)



| **Удалить шаблон** | **Del**

Удаление выбранного шаблона информационных сообщений.



| **Добавить параметр**

Добавление дополнительного параметра в шаблон информационных сообщений. [Добавление дополнительных параметров в шаблон](#)



| **Удалить параметр**

Удаление выбранного параметра из шаблона информационных сообщений. Допускается удаление только дополнительных параметров.

Копировать шаблон...


Создание копии выбранного шаблона информационных сообщений.

Переименовать шаблон... | F2

Изменение отображаемого имени выбранного шаблона информационных сообщений.

4.4.2. Создание шаблона информационных сообщений

Чтобы создать шаблон информационного сообщения, следует:

1. в окне **Шаблоны сообщений** в контекстном меню выбрать пункт **Новый шаблон**, либо нажать клавишу **Ins** или кнопку  на панели инструментов;
2. в появившемся окне ввести имя нового шаблона инф. сообщения и нажать **ОК**.

Примечание. В дальнейшем имя шаблона можно изменить с помощью клавиши **F2**.

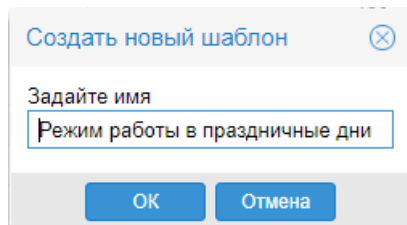


Рисунок 78. Ввод имени шаблона инф. сообщения

В результате будет создан пустой шаблон (в рабочей области будет заполнено поле **Имя шаблона**).

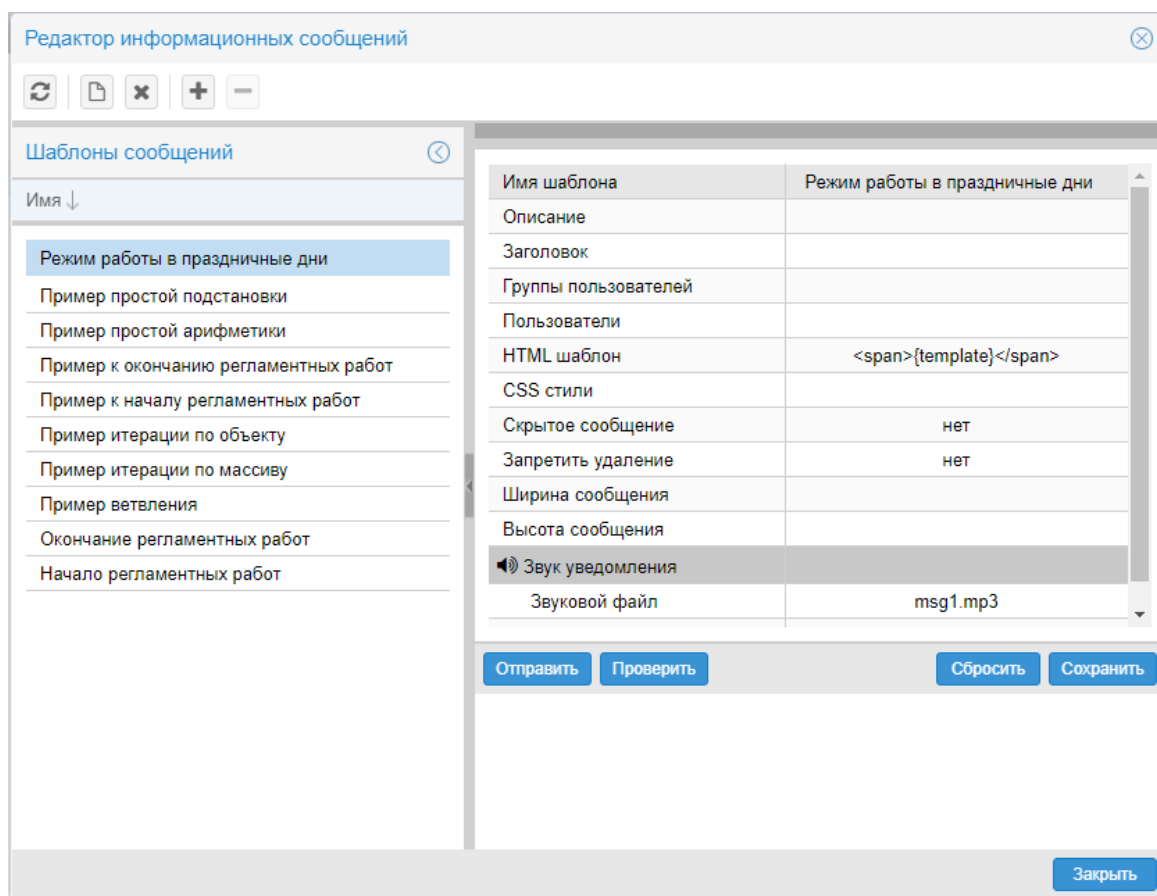


Рисунок 79. Создан пустой шаблон инф. сообщения

3. в рабочей области ввести параметры, определяющие содержимое сообщения и детали его отправки и хранения; см. подробности в разделах:
[Основные параметры шаблона информационных сообщений](#);
[Добавление дополнительных параметров в шаблон](#);
[Синтаксис HTML-шаблона](#);
4. по окончании ввода полей в рабочей области нажать кнопку **Сохранить**.

4.4.3. Основные параметры шаблона информационных сообщений

Далее в фигурных скобках (после наименования поля) указано имя параметра для использования в тексте html-шаблона.

Имя {template}

Уникальный идентификатор шаблона информационного сообщения.


Описание {description}

Описание сообщения, отображается в окне **Информационные сообщения**.

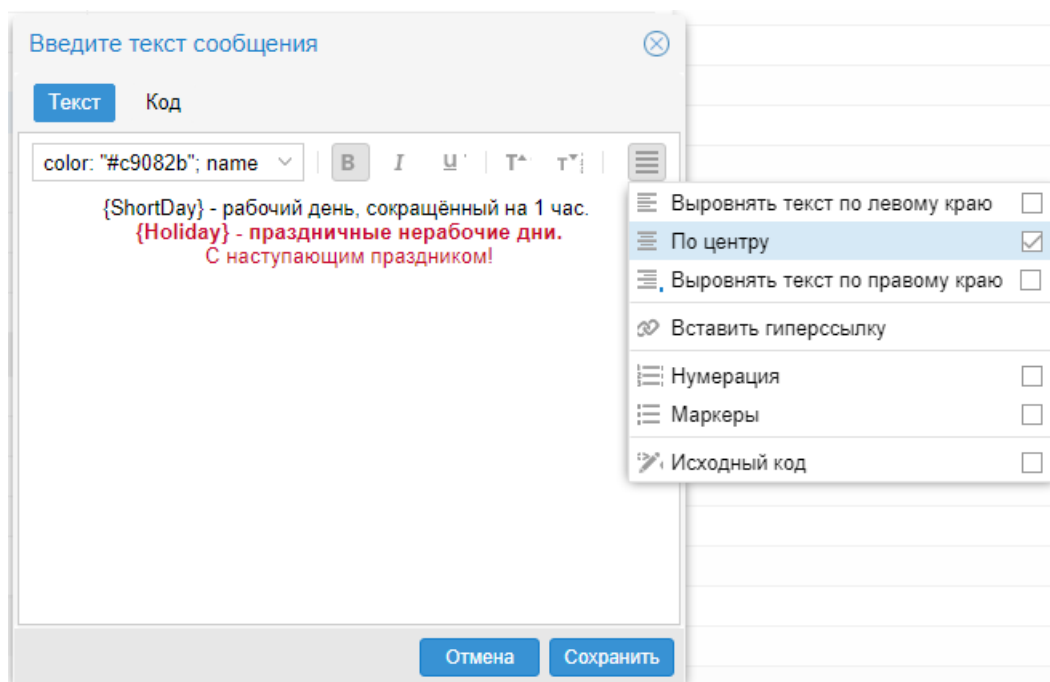
Заголовок {title}

Заголовок окна информационного сообщения; не отображается в окне **Информационные сообщения**.

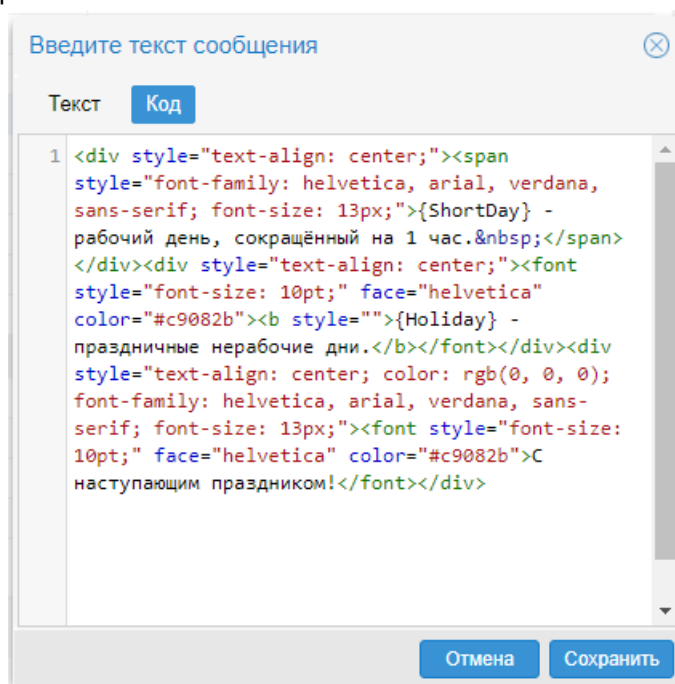
html-код сообщения

Html-шаблон информационного сообщения. Например, `<h1>{description}</h1>`. По нажатию кнопки  вызывается окно для ввода и форматирования содержимого сообщения:

на вкладке **Текст** можно вводить текст шаблона сообщения и использовать для форматирования кнопки на панели инструментов;



на вкладке **Код** введенный текст отображается в html-формате и также доступен для редактирования.



См. также [Синтаксис HTML-шаблона](#).

CSS-стили

CSS-стили, которые будут использованы в сообщении.

Группы пользователей

Список групп пользователей, которым будет отправлено информационное сообщение.

Пользователи

Список пользователей, которым будет отправлено информационное сообщение.

Таким образом, информационное сообщение отправляется пользователям из групп, указанных в поле **Группы пользователей**, а также пользователям, перечисленным в поле **Пользователи**.

Скрытое сообщение

Если флаг установлен (**true / да**), то сообщение не показывается получателю во всплывающем окне, а доступно для просмотра только в окне **Информационные сообщения**.

Запретить удаление (Safe)

Если флаг установлен (**true / да**), то сообщение нельзя удалить в окне **Информационные сообщения**. Чтобы вернуть возможность удаления сообщения, необходимо в шаблоне установить флаг **Запретить удаление** = **нет**.

Ширина сообщения

Ширина окна информационного сообщения в пикселах. По умолчанию формируется автоматически.



Высота сообщения

Высота окна информационного сообщения в пикселах. По умолчанию формируется автоматически.

Звук уведомления

Настройка звукового оповещения пользователя при получении им информационного сообщения.

Звуковой файл

Звуковой файл для воспроизведения можно выбрать из списка предустановленных файлов или загрузить с локального компьютера. С помощью кнопки  можно прослушать выбранный файл. Для сохранения выбранного файла в шаблоне сообщения нужно нажать кнопку .


Длительность звука

Длительность воспроизведения звукового файла в секундах.

4.4.4. Добавление дополнительных параметров в шаблон

Чтобы упростить и сократить внешний вид html-шаблона, можно создать дополнительные параметры и использовать их при написании текста html-шаблона.

Чтобы создать дополнительный параметр для шаблона инф. сообщения, следует:

1. в окне **Шаблоны сообщений** выбрать нужный шаблон;
2. нажать кнопку  на панели инструментов;
3. в появившемся окне ввести имя параметра (допустимы латинские буквы и цифры);

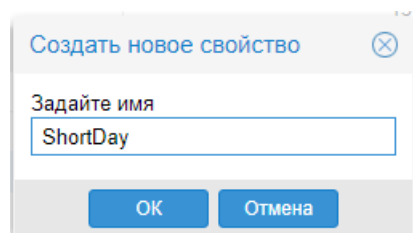


Рисунок 80. Ввод имени нового дополнительного параметра шаблона

4. в рабочей области ввести значение параметра;
5. нажать кнопку **Сохранить**.

Редактор информационных сообщений

Шаблоны сообщений

Имя ↓

- Режим работы в праздничные дни
- Пример простой подстановки
- Пример простой арифметики
- Пример к окончанию регламентных работ
- Пример к началу регламентных работ
- Пример итерации по объекту
- Пример итерации по массиву
- Пример ветвления
- Окончание регламентных работ
- Начало регламентных работ

| | |
|----------------------|--|
| Имя шаблона | Режим работы в праздничные дни |
| Описание | |
| Заголовок | Режим работы предприятия в праздники |
| Группы пользователей | Юг - Администраторы, Север - Администраторы |
| Пользователи | |
| HTML шаблон | <div style="text-align: left;"><span style="font-family: he... |
| CSS стили | |
| Скрытое сообщение | нет |
| Запретить удаление | нет |
| Ширина сообщения | 400 |
| Высота сообщения | 150 |
| Звук уведомления | |
| Звуковой файл | msg1.mp3 |
| Длительность звука | 1 |
| Holiday | 1, 2 мая |
| ShortDay | 30 апреля |

Отправить Проверить Сбросить Сохранить

ShortDay
не определено

Закрыть

Рисунок 81. Дополнительные параметры

Пример. Текст шаблона имеет вид:

```
<div><image src="/path1/path2/path3/image.png"/>...</div>
```

Можно создать дополнительный параметр IMG со значением /path1/path2/path3/image.png. В результате его использования исходный текст шаблона будет сокращён:

```
<div><image src="{IMG}"/>...</div>
```

4.4.5. Синтаксис HTML-шаблона

Полное описание синтаксиса html-шаблонов см. в документации [ExtJS](#).

В шаблонах можно использовать подстановки значений в html-код, условия, циклы и произвольный код JavaScript.

Простые подстановки

Значения параметров можно подставить с помощью простой подстановки, указав имя параметра внутри фигурных скобок:

```
{param_name}, {nested_param.field}
```

Итерация по массиву

Если значение параметра – массив (невозможно задать через диалог, но доступно через серверные методы), то можно задать отображение для каждого элемента массива с помощью особого тега <tpl> и атрибута for. Например:

```
<p>Целевые объекты:</p>
<ul>
  <tpl for="targets">
    <li><div class="{iconCls}"></div>{displayPath}</li>
  </tpl>
</ul>
```

Если параметр `targets` равен

```
[{"iconCls": "linux-computer", "displayPath": "/Tree/devel-lnx-28"},
 {"iconCls": "windows-computer", "displayPath": "/Tree/ws-069"}],
```

то приведенный выше шаблон выведет список из двух элементов.

Индекс текущего элемента доступен как `{#}`.

Текущий элемент (например, если это скаляр) доступен как `{.}`.

Также можно добавить произвольный текст между элементами массива:

```
<p>Объекты контроля: <tpl for="targets" between=", ">{displayPath}</tpl>.</p>
```

Итерация по объекту

Позволяет получить доступ к ключам и значениям произвольного объекта с помощью атрибута `foreach`, например:

```
<dl>
  <tpl foreach=".">
    <dt>{$}</dt> <!-- {$} содержит ключ свойства -->
    <dd>{.}</dd> <!-- {.} содержит значение свойства -->
  </tpl>
</dl>
```

Условные операторы

С помощью атрибутов `if`, `elseif` и `else` можно управлять отображаемой информацией. Например:

```
<p>Представлены следующие устройства:<p>
<ol>
  <tpl for="targets">
    <li>
      <tpl if="iconCls == 'linux-computer'">
        Компьютер Linux
      <tpl elseif="iconCls == 'windows-computer'">
        Компьютер Windows
      <tpl else>
        другие устройства
      </tpl>
    </li>
  </tpl>
</ol>
```

Также доступны атрибуты `switch`, `case` и `default`. Например:

```
<p>Представлены следующие устройства:<p>
<ol>
  <tpl for="targets">
    <li>
      <tpl switch="iconCls">
        <tpl case="linux-computer">
          Компьютер Linux
        <tpl case="windows-computer">
          Компьютер Windows
        <tpl default>
          Другие устройства
        </tpl>
      </tpl>
    </li>
  </tpl>
</ol>
```

В условиях доступны и другие операции сравнения, кроме равенства.

Важно: угловые скобки нужно указывать в форме html entity, например: `>`; вместо `>` и `<`; вместо `<`.

Арифметические операторы

В подстановках и условиях поддерживаются базовые арифметические операции над числовыми значениями: `+` (сложение), `-` (вычитание), `*` (умножение), `/` (деление).

Исполнение произвольного кода

Любой текст внутри `{ [...] }` считается кодом и выполняется внутри шаблона. Результат выполнения подставляется как значение. Важно, что внутри этих скобок доступ к переменным осуществляется через переменную `values`, а не напрямую. Например: вместо простого `{bar}` следует ввести `{[foo(values.bar)]}`, если к `bar` необходимо применить функцию `foo`. Также внутри цикла доступны переменные `xindex`, `xcount` и `xkey` для доступа к индексу элемента цикла (начиная с 1), числу элементов в коллекции, по которой идет цикл, и ключа текущего свойства, если применимо.

Также для использования доступны стандартные переменные, макросы, в зависимости от источника отправки информационного сообщения. См. [Параметры периода регламентных работ](#).

Также возможно выполнение кода без подстановки результата в шаблон, внутри блока `{% ... %}`.

4.4.5.1. Параметры периода регламентных работ

Для отображения в информационном сообщении сведений о регламентных работах можно использовать в html-шаблоне следующие дополнительные параметры:

- ❖ `{maintenance.targets}` – целевые объекты, на которых установлен период регламентных работ. Массив объектов, у каждого объекта есть свойства `path` (путь к объекту в дереве), `display_path`. Также могут быть доступны `target-type` (для целевых объектов) или `device` (для устройств);
- ❖ `{maintenance.reason}` – причина регламентных работ;
- ❖ `{maintenance.color}` – цвет штриховки графика в период регламентных работ. В HEX-формате без октоторпа впереди, например: `'ff0000'`, `'00ff00'`, `'0000ff'` – красный, зеленый и синий соответственно;
- ❖ `{maintenance.from}` – время начала периода регламентных работ в формате ISO 8601;
- ❖ `{maintenance.to}` – время окончания периода регламентных работ в формате ISO 8601;
- ❖ `{maintenance.user}` – информация о пользователе, который последним изменял период регламентных работ. По-умолчанию содержит свойства `id`, `displayName` и `title`;
- ❖ `{maintenance.extra}` – дополнительные данные, связанные с периодом регламентных работ. `null`, если дополнительных данных нет. Например, ссылка на нормативный документ: `{maintenance.extra.document_link}`.

4.4.5.2. Пример HTML-шаблона для инф. сообщения о регламентных работах

Ниже приведен пример текста HTML-шаблона для информационного сообщения о завершении регламентных работ. Во всплывающем окне информационное сообщение, сформированное по данному HTML-шаблону, будет иметь вид:

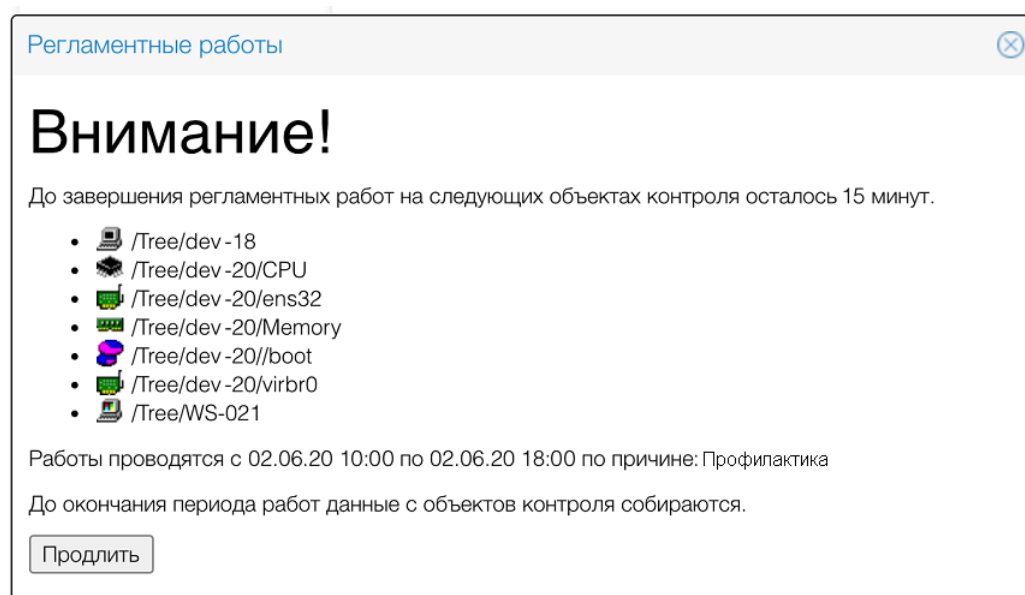


Рисунок 82. Пример информационного сообщения о завершении регламентных работ

```

<h1>Внимание!</h1>
<p>До завершения регламентных работ на следующих объектах контроля осталось
{[(moment(values.maintenance.time_end) - moment()) / 60 / 1000]} минут.</p>
<ul>
  <tpl for="maintenance.targets">
    <li>
      <div role="presentation" class="x-tree-icon x-tree-icon-custom x-
tree-icon-parent target-icon {[Core.mon.getDeviceIconCls(values)]}>
"></div>{display_path}
    </li>
  </tpl>
</ul>
<p>Работы проводятся с {[moment(values.maintenance.time_start).for-
mat(_T.DATE_FORMAT)]} по {[moment(values.maintenance.time_end).for-
mat(_T.DATE_FORMAT)]} по причине: {maintenance.reason}.</p>
<tpl if="maintenance.collect">
  <p>До окончания периода работ данные с объектов контроля собираются.</p>
<tpl else>
  <p>До окончания периода работ данные с объектов контроля не собира-
ются.</p>
</tpl>
<tpl if="Core.config.Runtime.config.auth.user == values.maintenance.user.id">
  <button onclick="alert('image in extended')">Продлить</button>
</tpl>

```

4.4.6. Тестирование шаблона информационных сообщений

Шаблон, в котором настроено отображение сообщения во всплывающем окне (**Скрытое сообщение = false / нет**), можно протестировать, т.е. просмотреть, в каком виде информационное сообщение будет доставляться пользователям. Для тестирования шаблона следует в окне **Шаблоны сообщений** выбрать нужный шаблон и нажать кнопку **Проверить**. В результате во всплывающем окне будет показано информационное сообщение, сформированное в соответствии с настроенным HTML-кодом.

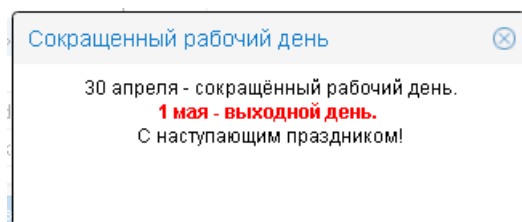


Рисунок 83. Тестирование информационного сообщения

4.4.7. Отправка информационного сообщения вручную

Чтобы отправить информационное сообщение, следует:

1. в окне **Шаблоны сообщений** выбрать нужный шаблон и в рабочей области нажать кнопку **Отправить**;
2. в появившемся окне при необходимости скорректировать текст сообщения и его форматирование (внесённые изменения не сохраняются в шаблоне);
3. нажать кнопку **Отправить**.

В результате информационное сообщение будет доставлено пользователям в соответствии с настройками шаблона.

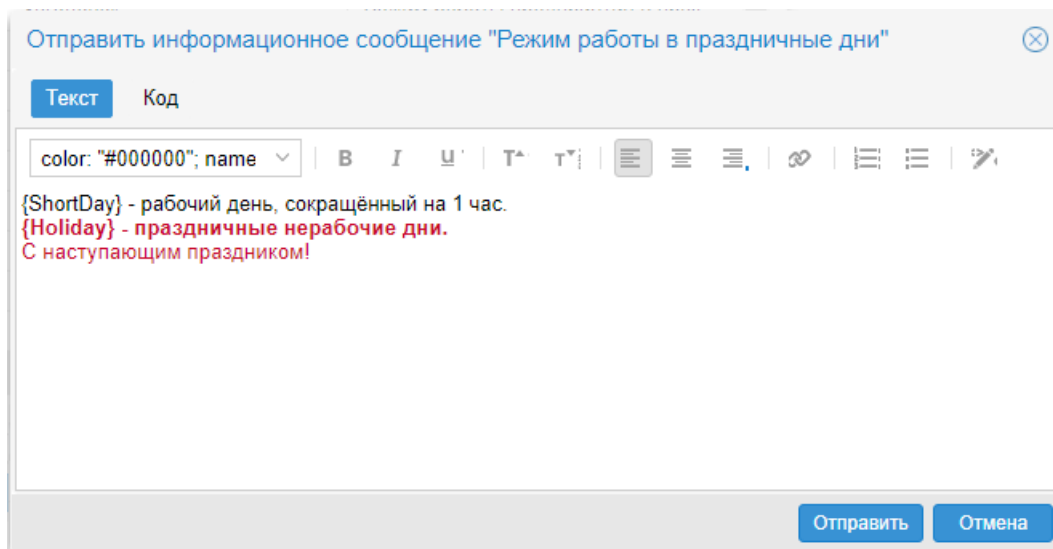


Рисунок 84. Отправка информационного сообщения

5. СПРАВОЧНИКИ

5.1. Системные процессы ПМ PhoenixDS Alarm Management

В настоящем справочнике приведено формализованное описание системных процессов ПМ **PhoenixDS Alarm Management**. Пункты отсортированы по именам процессов в алфавитном порядке.

Для управления системными процессами (если не указано особо) можно использовать утилиту `overlord.pl` или приложение **Диспетчер процессов** в Web-интерфейсе.

Параметры системных процессов можно условно разделить на две группы:

- ❖ стандартные – имеются у большинства системных процессов и описаны в п. Стандартные параметры системных процессов;
- ❖ специфические – описаны в пунктах, посвященных конкретным системным процессам.

5.1.1. Формирование сигналов (alerts)

Системный процесс `alerts`, основанный на процессе `alerts.pl`, обрабатывает данные, собранные системным процессом `collector`, и вычисляет статусы сигналов, фиксируя, таким образом, наличие проблем.

См. [Логическая схема вычисления статусов сигналов и отправки оповещений](#).

Параметры

`spooled`

Настройка высокопроизводительного режима обработки сигналов. При установленном флаге системный процесс `alerts` извлекает данные из буферных файлов, созданных системным процессом `collector` (при этом у системного процесса `collector` должен быть установлен флаг `spooler`).

`list`

Ветви дерева конфигурации (через запятую).

`spooldir`

Настройка высокопроизводительного режима обработки сигналов. Директория, из которой извлекаются собранные данные.

`reports`

Хранение статистики для ускорения создания отчетов.

`fifo`

Путь на директорию на сервере PhoenixDS и имя размещенного в ней файла, используемого системным процессом `omniorb-eventservice` при построении очереди событий. Параметр используется только в режиме высокопроизводительной обработки сигналов. Заданное значение должно совпадать со значением параметра `fifo` системного процесса `fnrmpunp`.

`qsize`

Размер очереди событий. Значение параметра следует задавать только при использовании высокопроизводительного режима обработки сигналов. По умолчанию параметр не задан.

`external-plug`

Для включения высокопроизводительного режима обработки сигналов следует ввести значение `FIFO`, а также задать параметры `fifo` и `qsize`.

По умолчанию значение не задано и высокопроизводительный режим не используется.

`no-esb-ds`

Получение данных из RRD-файлов. По умолчанию отключено (**false**).

`no-esb-alerts`

Параметр влияет на работу модуля **PhoenixDS Alarm Management** только при использовании сервисной шины ESB. Возможные значения:

- ❖ `false` (по умолчанию) – конфигурация сигналов хранится на сервере конфигураций, а значения сигналов – на сервере данных; системный процесс `alerts.pl` получает данные из шины ESB, и данные по изменению состояния сигнала передает обратно в ESB;
- ❖ `true` – конфигурация сигналов хранится в виде файлов на диске; системный процесс `alerts.pl` обменивается данными через шину ESB.

enabled-notifications

Шаблоны оповещений, обрабатываемые системным процессом `alerts`. Значение параметра – регулярное выражение для отбора шаблонов оповещений по имени. Оповещения по отобранным шаблонам будет формировать системный процесс `alerts` (вместо системного процесса `notifs`).

Начиная с версии 7.4 функционал формирования оповещений вынесен в отдельный системный процесс `notifs`. Поэтому изменять значение данного параметра не нужно.

balancing

Функция, ограничивающая множество сигналов, обрабатываемых данным системным процессом. Применяется для распределения нагрузки между несколькими системными процессами. Функция имеет вид: `Hash M/N`, где

`N` — общее количество системных процессов, которые обрабатывают одни и те же сигналы (согласно ветвям дерева конфигурации, указанным в параметре `list`, и параметру [Принадлежность](#) (`belongs-to`) набора сигналов);

`M` – порядковый номер системного процесса (число от 1 до `N`).

Системный процесс с параметром `--balancing 'Hash M/N'` обрабатывает сигналы по путям, которые удовлетворяют формуле: `hash(target_path) % N == M - 1`.

ignore_belongs_to

Если параметр установлен, то при определении множества обрабатываемых сигналов системный процесс игнорирует значение параметра [Принадлежность](#) (`belongs-to`), заданного в наборах сигналов. В этом случае системный процесс обрабатывает сигналы согласно ветвям дерева конфигурации, указанным в параметре `list`, и в соответствии с параметром `balancing`.

Распределение нагрузки между системными процессами `alerts` для оптимизации производительности ПП PhoenixDS

Первоначальные настройки ПМ **PhoenixDS Alarm Management** предусматривают один системный процесс на основе процесса `alerts.pl` – с именем `alerts`. С увеличением количества целевых объектов очереди к нему могут возрастать, что может отрицательно сказаться на производительности ПП PhoenixDS в целом.

```
[avalon@lnx-2 ~]$ /usr/local/lancelot/util/esb.queue.pl
-----
2019-05-23 15:57:20
S ALERTS 231152 +596
-----
```

Для решения данной проблемы рекомендуется создать несколько системных процессов на основе процесса `alerts.pl` и распределить нагрузку между ними.

При настройке системных процессов следует принимать во внимание значение поля [Принадлежность](#) (`belongs_to`) в наборах сигналов:

Если поле **Принадлежность** не заполнено, то обрабатывать сигналы данного набора могут системные процессы с именами вида `alertsN`, где `N` – необязательный порядковый номер. Например, системный процесс `alerts` или `alerts42` может обработать сигналы, а системный процесс `alerts-mega` – не сможет.

Аналогичный принцип соблюдается, если поле **Принадлежность** заполнено. Например, если в нем указан системный процесс `super_alerts`, то сигналы могут быть обработаны системными процессами с именами `super_alerts` и `super_alerts98`, и не могут быть обработаны системными процессами `alerts` и `alerts_super`.

Такую жесткую привязку набора сигналов к системному процессу с именем (шаблоном имени), указанным в поле **Принадлежность**, можно отключить, если для системного процесса на основе процесса `alerts.pl` установить параметр `ignore-belongs-to` (но обычно в этом нет необходимости).

Возможны следующие варианты:

- ❖ Создать несколько системных процессов с именами `alerts`, `alerts2`, `alerts3`, ... и настроить автоматическое распределение нагрузки с помощью параметра `balancing`. Например:

```
[avalon@lnx-2 ~]$ overlord.pl list alerts
Create new cHdb connection ...

alerts3(alerts.pl)
    balancing=Hash 3/3
    esb=127.0.0.1
    interval=210
    logfile=/usr/local/lancelot/var/lancelot-logs/alerts3.log
    no-esb-alerts=false
alerts2(alerts.pl)
    balancing=Hash 2/3
    esb=127.0.0.1
    interval=210
    logfile=/usr/local/lancelot/var/lancelot-logs/alerts2.log
    no-esb-alerts=false
alerts(alerts.pl)
    balancing=Hash 1/3
    esb=127.0.0.1
    interval=210
    logfile=/usr/local/lancelot/var/lancelot-logs/alerts.log
    no-esb-alerts=false
```

- ❖ В дополнение к системному процессу `alerts` создать еще несколько – под разными именами, например `alerts-param1`, `alerts-param2`. Чтобы распределить нагрузку, в нескольких наборах сигналов задать параметр [Принадлежность](#) (`belongs-to`), указав имя одного из созданных системных процессов.
- ❖ Создать несколько системных процессов с разными именами, например `alerts1`, `alerts2`, а для распределения нагрузки в параметре `list` указать для разных системных процессов различные ветви дерева конфигурации (`--list=/Sub-tree1`, `--list=/Sub-tree2`). Таким образом, необходимо разделить дерево конфигурации между системными процессами на основе `alerts.pl`. Например:

```
[avalon@qa-lnx-2 ~]$ overlord.pl list alerts
Create new cHdb connection ...

alerts2(alerts.pl)
    disabled=false
    esb=127.0.0.1
    interval=210
    list=/Ros/South/Objects,/Ros/North/Objects
    logfile=/usr/local/lancelot/var/lancelot-logs/alerts-param-2.log
    no-esb-alerts=false
alerts1(alerts.pl)
    disabled=false
    esb=127.0.0.1
    interval=210
    list=/Ros/East/Objects,/Ros/West/Objects
    logfile=/usr/local/lancelot/var/lancelot-logs/alerts-param-1.log
    no-esb-alerts=false
```

5.1.2. Формирование оповещений (notifier)

Системный процесс `notifs`, основанный на процессе `notifier.py`, в соответствии с заданными администратором настройками формирует оповещения о возникшей проблеме одним или несколькими способами: email, sms, прерывание snmp или запись в лог.

Параметры

`digest-interval`

Периодичность отправки дайджестов по электронной почте (используется только методом оповещений Email). Значение параметра – интервал в секундах.

`disable`

Шаблоны оповещений, игнорируемые системным процессом `notifs`. Значение параметра – регулярное выражение для отбора шаблонов оповещений по имени. Оповещения по отобранным шаблонам формироваться не будут.