

Blocks[®]

версия 3.1

© Все права защищены, PIXILAB Technologies AB, Все права защищены.

Торговые марки: PIXILAB Blocks, PIXILAB Moments (PIXILAB Technologies AB), Google Chrome (Google, Inc), Windows (Microsoft, Inc), macOS (Apple, Inc), Brightsign (Brightsign, LLC), Dataton WATCHOUT (Dataton AB), Instagram (Instagram, LLC).
Перевод INTMEDIA www.intmedia.ru.

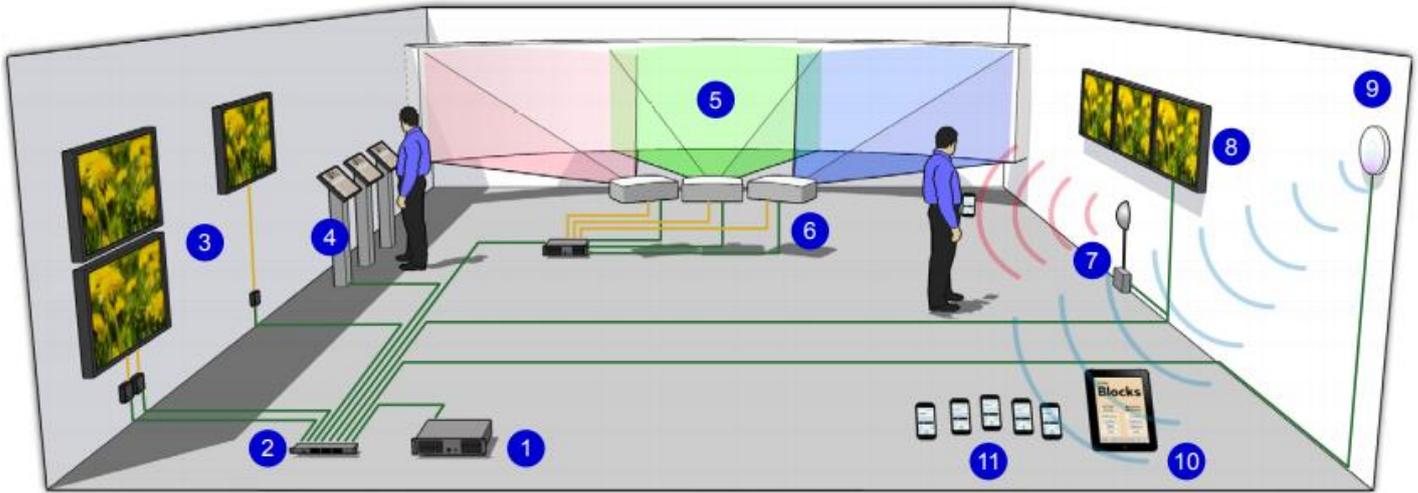
СОДЕРЖАНИЕ

1. Знакомство с Blocks	4
2. НАЧАЛО РАБОТЫ	6
Mac	6
Windows	9
Linux	10
3. ОСНОВНЫЕ ПОНЯТИЯ	12
Спот (spot)	12
Блоки	12
Управляемые подсистемы	13
Задачи (Tasks)	14
4. СПОТЫ	15
Настройки спота (Spot Settings)	16
Дисплей-спот	16
Visitor (гость, посетитель)	19
Location (местоположение)	21
WATCHOUT	21
Группа спотов	22
5. БЛОКИ	23
Слайд-шоу	26
Composition (композиция)	29
Book (книга)	31
Stack (штабель)	31
Scroller (прокрутка)	32
Attractor	33
Tag Selector (селектор тэгов)	34
Media File (медиафайл)	34
Text	36
Locator (локатор)	39
Synchronizer (синхронизатор)	40
Блок Web	42
Widget	43
Reference (ссылка)	43
Live Video	44
Instagram	45
Schedule (расписание)	46
6. ПАНЕЛИ УПРАВЛЕНИЯ	48
Button (кнопка)	49
Indicator (индикатор)	53
Slider (ползунок)	53
Bar (полоса)	53
Текст	53
Text Input (ввод текста)	54
7. MANAGE (УПРАВЛЕНИЕ)	55
Users (пользователи)	55
Licenses (лицензии)	55
Кластеры WATCHOUT	56
Instagram Feeds (загрузка из Instagram)	57
Модули I/O Modbus	59
Сетевые устройства	61
Свет: Art-Net / DMX-512	63
8. TASK (ЗАДАЧА)	65
Realm (область)	66
Группа (Group)	67

Task (задача)	68
Операторы	71
9. СВОЙСТВА И ФУНКЦИИ	80
Дисплей-спот (Display Spot)	83
Visitor Spot (гостевой спот)	84
Location (местоположение)	84
Задача (Task)	84
Канал Modbus	85
Art-Net/DMX	85
WATCHOUT	86
Сеть (Network)	88
10. АВТОРИЗАЦИЯ И РОЛИ	88
Настраиваемые ограничения доступа	90
A. ФАЙЛОВАЯ СТРУКТУРА СЕРВЕРА	94
B. ДОПОЛНИТЕЛЬНЫЕ СКРИПТЫ И ДРАЙВЕРЫ на TypeScript	95

1. Знакомство с Blocks

PIXILAB BLOCKS® - программное обеспечение для создания мультимедийных сред музеев, выставок, общественных пространств и корпоративных мероприятий. ПО BLOCKS – это сочетание управления контентом и дисплеями, функций системы управления, интерактивности и возможностей мобильного гида.



Система на базе Blocks и схема подключений. Сеть (зеленый), видео (оранжевый), Wi-Fi (синий) ИК / метки (красный).

На приведенной выше иллюстрации показана типичная система Blocks, содержащая следующие важные элементы (они помечены цифрами в синих кружочках).

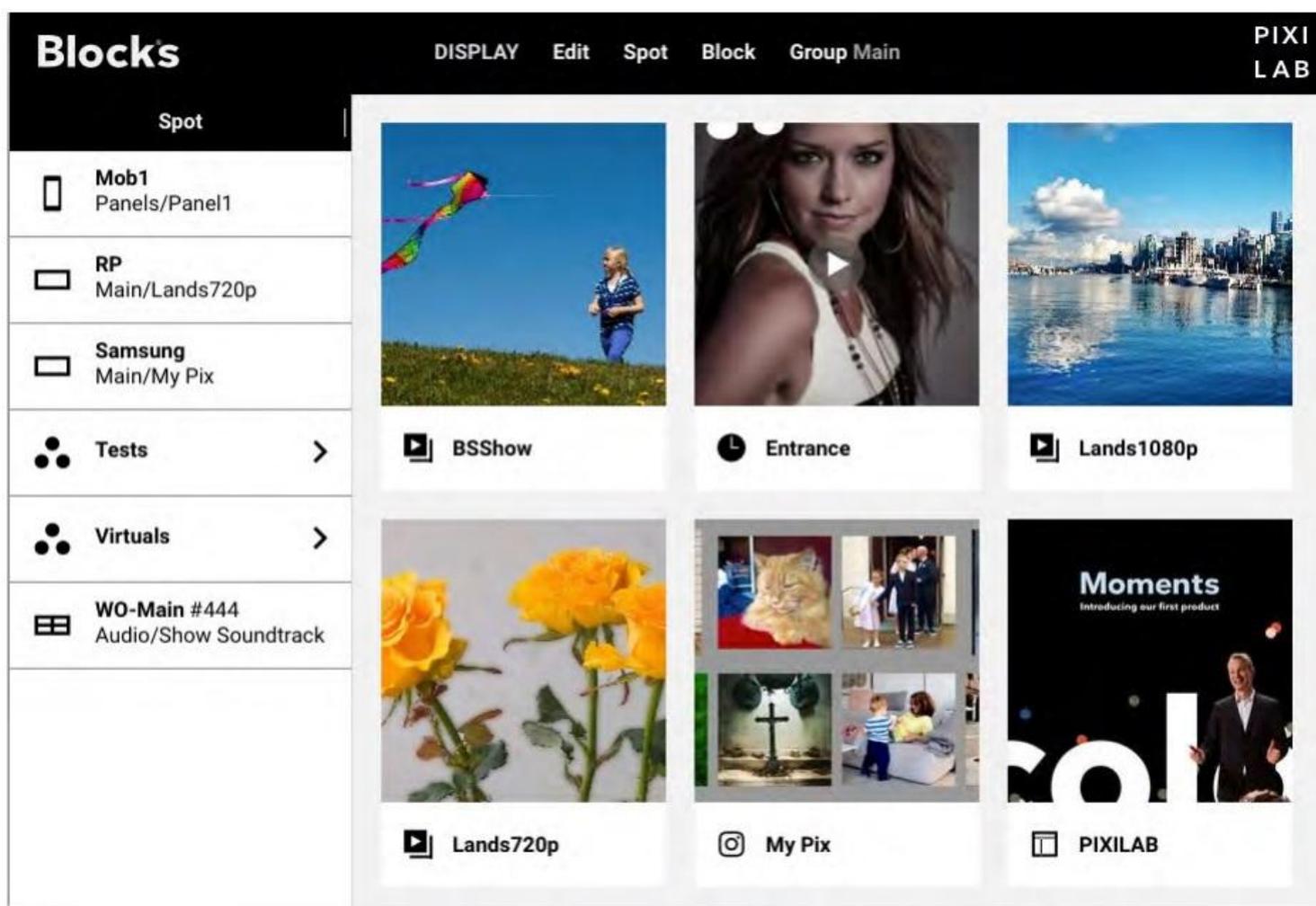
1. Сервер Blocks.
2. Проводная сеть.
3. Дисплеи с отдельными плеерами.
4. Интерактивные сенсорные киоски.
5. Подсистема Dataton WATCHOUT® для проекции на вогнутый экран.
6. Проекторы и другие устройства, контролируемые через сеть.
7. Реле, переключатели и датчики (здесь - ИК-датчик движения), подключаются через стандартные интерфейсы.
8. Интеллектуальные дисплеи со встроенными плеерами подключают непосредственно к сети.
9. Сеть Wi-Fi для внутреннего управления и/или общего доступа.
10. Панели управления для персонала.
11. Смартфоны пользователей или арендуемые смартфоны для использования в качестве мобильных гидов.

Сервер

В основе BLOCKS лежит серверное программное обеспечение. Эта программа может работать на вашем персональном компьютере или на мощном специализированном серверном оборудовании. Персональный Mac или Windows PC удобно использовать на стадии продакшн и для проведения быстрых тестов, а также для запуска небольших проектов. Для более масштабных и сложных инсталляций необходимо специальное серверное оборудование, работающее под ОС Linux.

Редактор

Вы создаёте и редактируете презентации с помощью обычного веб-браузера, который подключает к серверу Blocks по сети. Здесь контент (например, изображения, видео и звук) объединяется в блоки. Затем контент назначается дисплеям или мобильным устройствам, которые мы будем называть спотами (*spots*).



На странице Display редактора видны дисплеи и другие споты, а также блоки контента.

Споты (Spots)

Точки интереса, такие как дисплеи, кинотеатры, локации на местности или внутри зданий, обычно называются спотами. Споты указаны в левой части страницы Дисплей (см. выше) и могут быть собраны в группы и подгруппы для упрощения навигации и программирования.

Спотами будем считать и мобильные устройства, которые могут быть постоянной частью системы (например, iPad в качестве панелей управления) или временной (например, смартфоны посетителей).

Блоки (Blocks)

Подобно кирпичикам LEGO, блоки можно комбинировать практически как угодно. Это позволяет готовить контент для любых случаев - от простого слайд-шоу в стиле цифровых вывесок до интерактивного киоска или системы мобильного гида, где видео и звук воспроизводятся синхронно на нескольких устройствах.

Например, используйте блок композиция (Composition), комбинируя изображения и кнопки для создания пользовательского интерфейса. Если ваш проект сложный, используйте блок Книга (Book) для размещения нескольких блоков композиций (Composition) на нескольких страницах. Чтобы рассказать историю, используйте блок слайд-шоу (Slide Show), собрав ряд изображений или коротких видеоклипов. Для привлечения посетителей, например, к интерактивному киоску можно использовать блок Attractor.

2. НАЧАЛО РАБОТЫ

В этой главе описывается, как создавать и запускать Blocks для создания, тестирования, а также для запуска готовой инсталляции. Вы можете использовать ноутбук Mac или Windows на стадии продакшн и базовых тестов всех функций Blocks, но для запуска готовой инсталляции мы рекомендуем использовать выделенный сервер на базе Linux.

Mac

Использование ноутбука или настольного компьютера удобно для работы со сторонним программным обеспечением для создания контента (изображений, видео и звука). Хотя вы можете получить доступ к серверу Blocks с любого компьютера через локальную сеть, наличие приложения Blocks на вашем ноутбуке означает, что вы можете работать и над созданием контента, и тестировать споты (spots).

Инсталляция

Чтобы установить приложение Blocks на вашем компьютере Mac, выполните следующие действия:

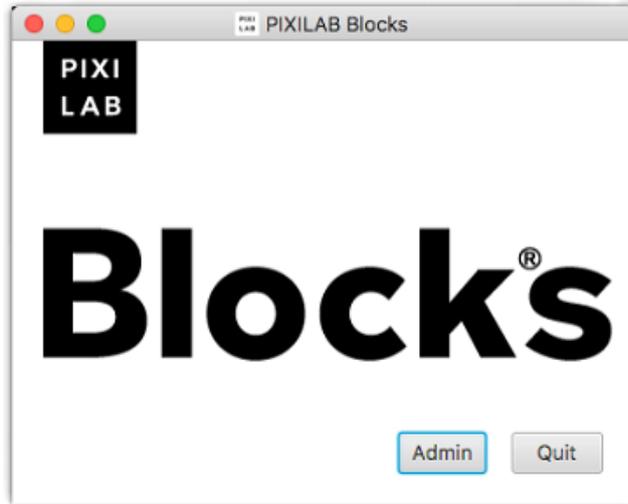
- Загрузите Мастер-установщик PIXILAB_Blocks.dmg из <https://int.pixilab.se/docs/blocks>.
- Откройте файл DMG, прочтите и подтвердите согласие с лицензионным соглашением.
- После открытия образа диска вы увидите окно, подобное показанному на рисунке ниже.
- Если вы впервые устанавливаете Blocks на данном компьютере, запустите установщик CmInstall.pkg. В результате будет установлен драйвер лицензионного ключа. После установки драйвера необходимо перезагрузить Mac. Не нужно повторять этот шаг при обновлении Blocks до более свежей версии.
- Перетащите значок приложения PIXILAB Blocks в папку «Приложения» (Applications) или в другое подходящее место.
- Подключите лицензионный ключ PIXILAB к порту USB на вашем компьютере.



Окно Мастера установки для Mac после открытия файла PIXILAB_Blocks.dmg.

Запуск сервера Blocks

Найдите приложение «PIXILAB Blocks» в папке «Приложения» (Applications) или там, где вы его оставили. Чтобы упростить использование в будущем, можно перетащить значок приложения в Dock. Затем запустите приложение Blocks. Откроется окно, показанное на рисунке ниже. Поскольку всё администрирование сервера выполняется по сети, в этом окне мало информации. Откройте редактор Blocks (Blocks Editor), нажав кнопку «Администратор» (Admin).



Окно сервера PIXILAB Blocks.

Доступ к редактору Blocks

Откройте Редактор Blocks, нажав кнопку «Администратор» (Admin) в окне сервера или перейдя по адресу сервера, за которым следует `/edit`. Например, если ваш Mac имеет IP-адрес 10.0.2.10, вы сможете получить доступ к редактору, введя следующее в свой веб-браузер:

<http://10.0.2.10:8080/edit>

Появится запрос на имя пользователя и пароль. При первом запуске сервера есть только один пользователь с именем «admin» и с паролем «rīxī». Вы должны как можно скорее изменить этот пароль, если сервер общедоступен (это делается на странице «Управление» (Manage), как описано в разделе «Пользователи» (“Users”).

После входа в систему, вы увидите окно, похожее на окно, показанное выше в разделе «[Редактор](#)».

▲ Редактору Blocks необходим современный веб-браузер, такой как Google Chrome, Apple Safari, Firefox или Microsoft Edge.

Добавление дисплея

Для проверки контента, который вы создаёте в приложении Blocks, вам нужно отображать его на Display Spot. Это может быть просто компьютер с современным веб-браузером, например Google Chrome или Apple Safari. Желательно, чтобы браузер работал в полноэкранном режиме, а адресная строка и другие элементы управления были скрыты. Пусть ваш Mac-сервер имеет IP-адрес 10.0.2.10, как в приведенном выше примере, назначьте ваш дисплей-компьютер на:

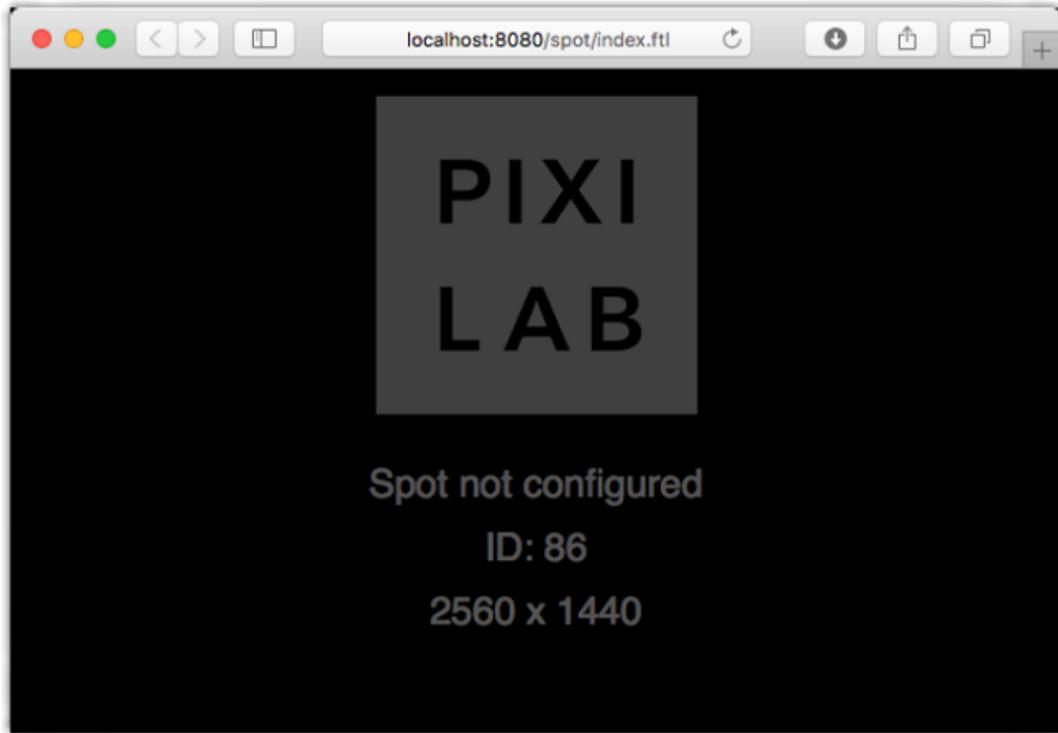
[http:// 10.0.2.10:8080/spot](http://10.0.2.10:8080/spot)

Если у вас нет отдельного компьютера, вы можете просто открыть ещё одно окно браузера на своем ноутбуке, указав его по следующему адресу:

<http://localhost:8080/spot>

▲ Имя `localhost` всегда ссылается на IP-адрес локального компьютера, поэтому его можно использовать, как сокращенное имя вместо IP-адреса компьютера

Вы увидите чёрное окно, как показано ниже. В этом окне будет отображаться идентификационный номер (ID) спота, а также его полноэкранные размеры.



Окно Display Spot перед добавлением на сервер.

Оставив окно Display Spot на экране, вернитесь к редактору Blocks. Выберите “Add Display Spot” (добавить дисплей-спот) в меню Spot. Выберите идентификационный номер, отображаемый на дисплее, как идентификатор спота (Spot ID) и введите имя в поле Name.

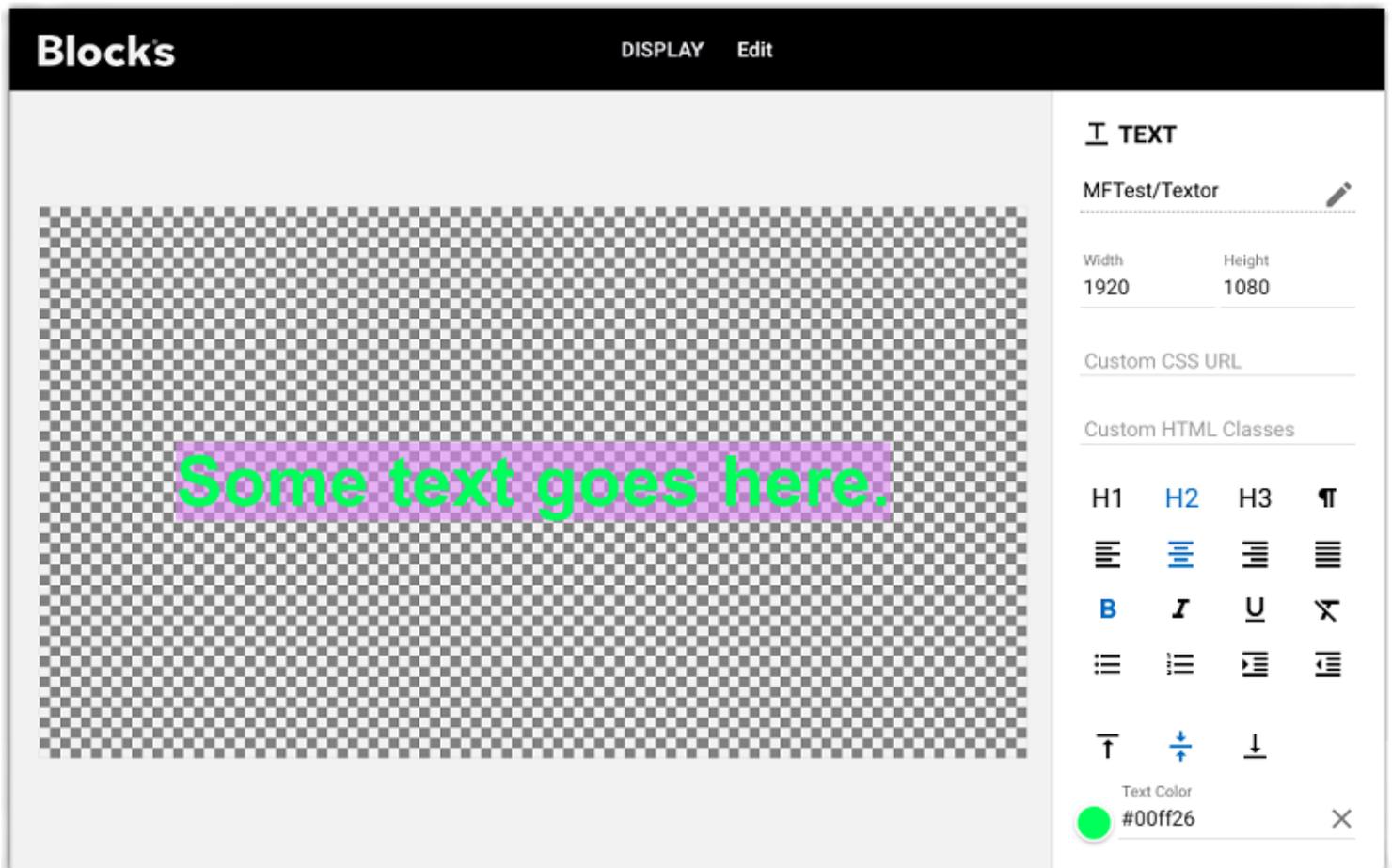
Добавление Display Spot через назначение Spot ID.

При нажатии ОК в диалоговом окне появится имя спота вместо “Spot not configured” (спот не сконфигурирован), это значит, что дисплей успешно принят в систему.

Отображение блока

Чтобы что-то появилось на дисплее, достаточно просто перетащить блок в спот на странице [Display](#). Если в вашей системе ещё нет блоков, вы можете создать простой текстовый блок следующим образом:

- Выберите “Add Text” (добавить текст) в меню Block, укажите имя блока и нажмите «ОК».
- Дважды щёлкните на текст, чтобы отредактировать его, центрируйте его по горизонтали и вертикали, используя соответствующие кнопки, расположенные справа.
- Измените цвет текста на белый.
- Вернитесь на главную страницу (Display page), щёлкнув на логотип “PIXIlab Blocks” в верхнем левом углу.
- Перетащите созданный текстовый блок на спот, чтобы он отобразился на дисплее.



Создание текстового блока.

Windows

Использование ноутбука или настольного компьютера удобно для работы со сторонним программным обеспечением для создания контента (изображений, видео и звука). Хотя вы можете получить доступ к серверу Blocks с любого компьютера через локальную сеть, наличие приложения Blocks на вашем ноутбуке означает, что вы можете работать и над созданием контента, и тестировать споты (spots).

Инсталляция

Чтобы установить Blocks на ваш компьютер с ОС Windows, выполните следующие действия:

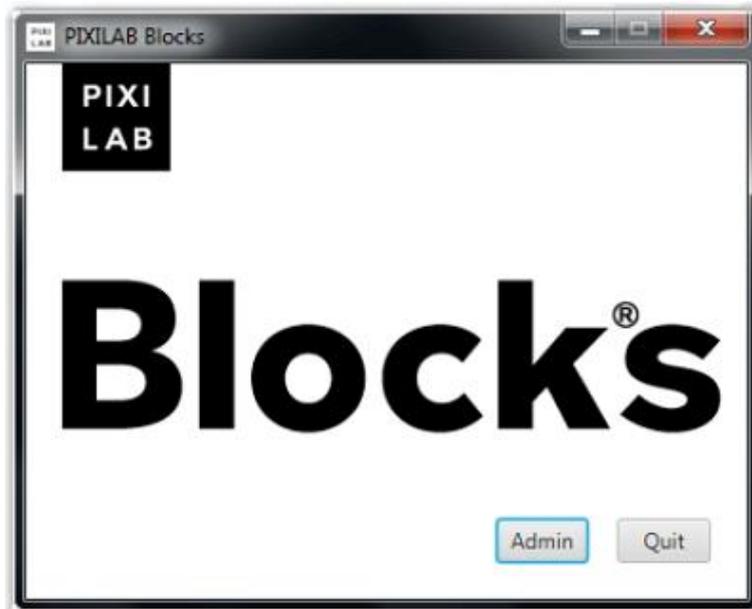
- Загрузите Мастер установки PIXILAB_Blocks.zip с: <https://int.pixilab.se/docs/blocks>.

- Щёлкните правой кнопкой мыши на файл PIXILAB_Blocks.zip и выберите “Extract All” (извлечь все), чтобы распаковать ZIP-архив.
- Откроется окно с двумя значками, показанными справа.
- Если вы устанавливаете приложение Blocks на этот компьютер впервые, запустите Мастер установки “CodeMeter Runtime.exe”. При этом будет установлен драйвер лицензионного ключа. После установки драйвера следует перезагрузить компьютер. Вам не нужно повторять этот шаг при обновлении приложения Blocks до более новой версии.
- Дважды щёлкните на “PIXILAB-Blocks-Installer.exe” и следуйте инструкциям по установке приложения Blocks.
- Подключите лицензионный ключ PIXILAB к порту USB на вашем компьютере.



Запуск сервера Blocks

Щёлкните на “PIXILAB Blocks Server” в меню Start (Пуск). Откроется окно, показанное ниже. Поскольку всё администрирование сервера выполняется по сети, в этом окне мало информации. Вы можете открыть редактор Blocks (Blocks Editor), нажав кнопку Admin. Затем выполните действия, описанные ранее в этой главе в разделе «[Доступ к редактору](#)».



Окно сервера PIXILAB Blocks.

Linux

Для постоянных инсталляций рекомендуется использовать сервер на Linux. В отличие от версий для Mac и Windows, которые могут быть установлены на вашем личном компьютере, версия приложения Blocks для Linux предназначена для специального использования. Это гарантирует наилучшую долговременную стабильность, особенно в сочетании с проверенным оборудованием. Версия Blocks для Linux предоставляется в следующих двух видах:

- Предварительно установленная на серверное оборудование, предоставляемое PIXILAB или одним из наших авторизованных посредников.
- В качестве предварительно настроенного “образа Linux” (Linux image) для установки на подходящее оборудование.

Первый вариант предоставляет вам укомплектованную под ключ комбинацию оборудования/программного обеспечения, готовую к использованию оптимизированную систему. Это сервер в 19-дюймовом корпусе для монтажа в стойку 3U. Базовая конфигурация имеет два сетевых порта по 1 Гб, но она также может быть снабжена картой с двумя портами 10 Гб. Свяжитесь с PIXILAB для получения более подробной информации о технических характеристиках и доступных опциях.

Опция “образ Linux” позволяет загружать и устанавливать предварительно настроенный сервер на ваше оборудование. Он протестирован и способен гарантированно работать на компьютерах малого форм-фактора Intel NUC поколения 7 и 8 с SSD-диском M.2 или SATA (рекомендуем M.2 для лучшей производительности).

▲ Хотя “образ Linux”, скорее всего, будет работать и на другом оборудовании, его пригодность, надежность и производительность на таком оборудовании отнюдь не гарантирована.

На [PIXILAB Wiki](#) можно найти инструкции, как получить и установить образ Linux.

В дополнение к настроенному ядру сервера Linux, аппаратное обеспечение/образ сервера Linux имеет следующие предустановленные и настроенные функции:

- Программное обеспечение PIXILAB Blocks Server и связанные с ним дополнительные компоненты и конфигурацию.
- Программа SSH для безопасного удаленного доступа.
- Поддержка загрузки по сети PIXILAB Players (PXE).
- Серверы DNS и DHCP.
- Сетевой контроллер Wi-Fi промышленного класса, включая необходимое хранилище данных.
- Программное обеспечение для удаленного управления.
- Средства разработки и библиотеки для драйверов устройств и сложных сценариев.
- Программное обеспечение для устранения неисправностей и диагностики сети.
- Artnet / DMX-512 для тестирования программного обеспечения.
- Сетевая совместимость Windows / Mac (SMB).
- Совместимость с USB-накопителем Windows / Mac (с использованием файловой системы ExFAT).

Все перечисленные выше функции установлены и настроены должным образом, но большинство дополнительных функций отключены по умолчанию из соображений безопасности. Инструкции по их включению, а также перечисление связанных с этим последствий для безопасности можно найти в разделе “Linux-based Server” (Сервер на базе Linux) по адресу https://int.pixilab.se/docs/blocks#linux-based_server.

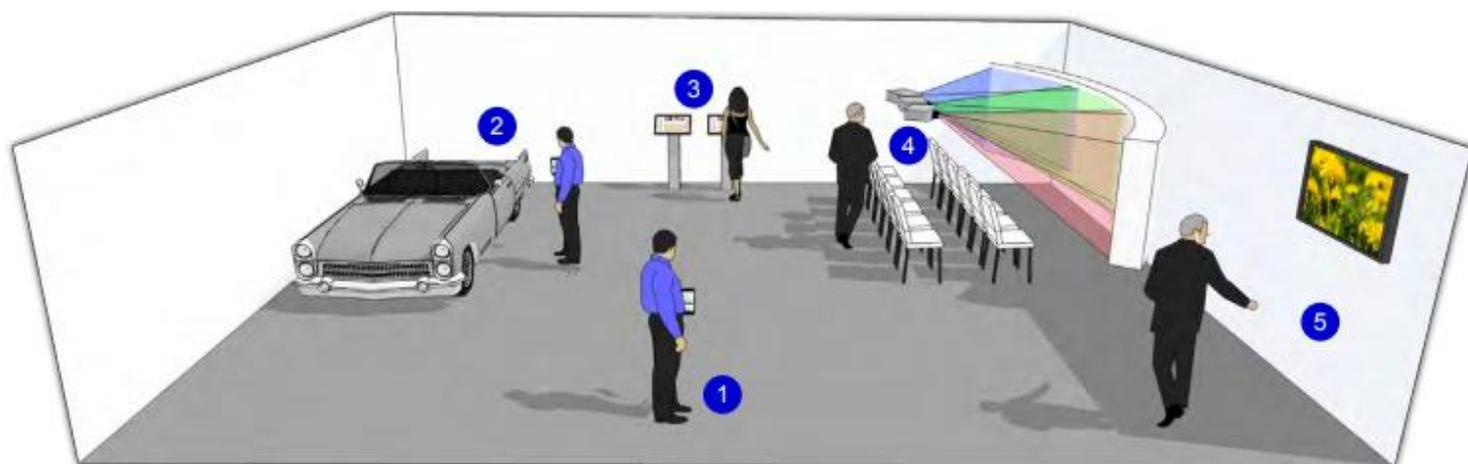
▲ “Linux Server License” (лицензия для сервера Linux) нужна для запуска сервера под Linux (см. раздел «[Лицензии](#)»). Это не включено в версию “Personal License” для Blocks.

3. ОСНОВНЫЕ ПОНЯТИЯ

В этой главе дается описание понятий, используемых в Blocks, таких как споты, блоки и tasks (задачи). Эти термины подробно рассматриваются в следующих главах.

Спот (spot)

В самом общем смысле spot - это место. Это может быть комната, объект на местности или внутри здания. В контексте Blocks спот, как правило, связан с некоторыми медиа, такими как звук или видео, привязанными к этому споту. Хотя спот может быть просто некоторой точкой в пространстве, он также часто привязан к какому-то оборудованию: дисплею, сенсорному киоску или проекционной системе с широкоэкранным шоу WATCHOUT.



Споты внутри помещения.

Ссылаясь на приведенный выше рисунок, перечислим типы спотов, принятых в Blocks:

1. **Гостевой спот (Visitor spot)**: персональные мобильные устройства посетителей.
2. **Локация (Location)**: объект или местоположение, в данном случае – это автомобиль.
3. Сенсорный киоск на базе **дисплей-спота (Display Spot)** с интерактивным блоком (контентом).
4. **WATCHOUT спот**: кинотеатр с широким экраном, где WATCHOUT Spot обеспечивает синхронизированный звук через мобильные устройства посетителей.
5. **Дисплей-спот (Display Spot)**: воспроизведение видео, изображений и другого контента, созданного в приложении Blocks. Кроме того, синхронизированный звук на нескольких языках может быть предоставлен через мобильные устройства посетителей.

См. главу "[Споты](#)" для получения подробной информации о типах спотов.

Блоки

В приложении Blocks материалы, которые вы используете для создания контента (и это вносит путаницу), называются блоками. Подобно кирпичикам LEGO, эти блоки можно комбинировать как угодно для получения разнообразных презентаций. Споты соотносятся с физическими местоположениями или дисплеями, блоки же более абстрактны, и чтобы освоить работу с ними, потребуется немного больше времени.

Хотя некоторые блоки могут использоваться сами по себе (например, можно использовать только текстовый блок для отображения текстового сообщения), блоки обычно объединяют разными способами. Это получается путём размещения блоков внутри других блоков, что напоминает устройство матрёшки.

Вот некоторые из типов блоков:

- **Слайд-шоу** для показа последовательности изображений, видео или другого контента.
- **Книга (Book)** для создания интерактивных презентаций с использованием нескольких страниц.
- **Композиция (Composition)** для размещения других блоков или интерактивных элементов на странице.
- **Скроллер (Scroller)** для интерактивной прокрутки контента по горизонтали или вертикали.
- **Аттрактор (Attractor)** для отображения цикла привлечения посетителя вплоть до момента его взаимодействия со спотом и переключением на активный контент этого спота.
- **Синхронизатор (Synchronizer)**, синхронизирует воспроизведение на разных спотах видео и звука. Например, звук идёт через смартфон синхронно с видео, отображаемым на дисплее.

Кнопки, движки и другие элементы интерактивного управления используются наряду с другими блоками. Но такие элементы управления можно использовать только внутри блока **Composition**.

В главе «[Блоки](#)» подробно описывается каждый тип блоков.

Управляемые подсистемы

Споты и блоки обеспечивают создание контента, его отображение и интерактивность, но приложение PIXILAB Blocks может управлять и многими другими вещами, которые можно добавить и настроить на странице «*Manage*» (*Управление*).

The screenshot shows the 'Blocks' management interface. At the top, there are buttons for 'MANAGE', 'Edit', and 'Add'. Below is a table with columns: 'Module', 'IP Address', 'Enabled', and 'Notes'. The 'Modbus I/O Modules' section is expanded, showing a table with columns: 'Channel', 'Direction', 'Type', and 'Alias'. The table lists several modules, with the third one (Channel 3, Direction Out, Type Digital, Alias theaterDoor) highlighted in blue. To the right of the table, there are configuration options for the selected module, including 'Module Name *' (MOXA), 'IP Address *' (10.0.1.113), 'Enable Module' (checked), 'Note' (Cabinet 2), 'Modbus Channel Number *' (3), 'Alias (under 'IO')' (theaterDoor), 'Signal Direction' (Output selected), and 'Signal Type' (Digital selected).

Module	IP Address	Enabled	Notes
MOXA	10.0.1.113	YES	Cabinet 2

Channel	Direction	Type	Alias
1	Out	Digital	cleaningLights
2	Out	Digital	curtain
3	Out	Digital	theaterDoor
4	Out	Digital	DALI_5
10001	In	Digital	motion1
10002	In	Digital	motion2

На приведенном выше рисунке показано, что можно добавить модули ввода/вывода, поддерживающие протокол Modbus для управления и взаимодействия с контактными разъемами и датчиками. Modbus – это стандартный протокол, поддерживаемый большинством поставщиков. Таким же образом вы можете управлять освещением (через Art-net / DMX512), сетевыми устройствами, такими как проекторы (включение / выключение питания, переключение входов), внешние источники данных (например, считыватель штрих-кода) и т. д. Для взаимодействия с такими сетевыми устройствами доступно специализированное программное обеспечение ([драйверы устройств](#)). Можно контролировать кластеры WATCHOUT для интеграции в систему Blocks крупномасштабных систем отображения.

Задачи (Tasks)

Многие основные функции управления можно реализовать прямым соединением кнопок с объектами управления. Такое однозначное соединение «один-к-одному» выполняется в редакторе блока Композиция (Composition) при добавлении и размещении кнопок. В качестве бонуса обеспечивается обратная связь: на кнопке отображается статус устройства. Всё это происходит без какого-либо программирования с вашей стороны.

Однако некоторые функции не так просты, как однозначное взаимодействие кнопки и функции устройства. Например, метод прямой связи можно использовать для включения/выключения одного проектора, но этот метод не годится, если вы хотите управлять тремя проекторами, а также уровнем освещения в комнате с помощью одной кнопки. Здесь потребуется управление множественными режимами.

Более того, прямая связь – это просто прямая связь. Нет никаких «если» (*if*) и «но» (*but*) или других вариантов. Иногда вы хотите, чтобы всё происходило только при определенных условиях; например, не в выходные дни, или когда в комнате не более 10 человек. Вы можете захотеть, чтобы некие события происходили в зависимости от какого-то состояния.

Все это и многое другое можно предусмотреть на странице *Задачи (Task)*. Эта страница в приложении Blocks предоставляет базовую, но удивительно мощную среду программирования. Так же, как вы совмещаете блоки контента на странице Display, на странице «Задача» (Task) можно подключить команды программирования простым перетаскиванием. Выберите в раскрывающемся меню свойства, которыми нужно управлять, так же, как вы это делали при назначении кнопок. Затем используйте простой язык выражения, чтобы детально определить, что и как нужно сделать. Язык выражений основан на JavaScript; самом популярном языке программирования, и *lingua franca* в Интернете.

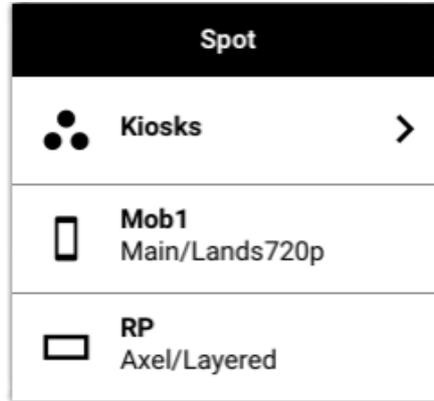
The screenshot shows the 'TASK' configuration interface in the Blocks application. At the top, there's a header with 'Blocks' and 'TASK Edit Realm Main'. Below the header, there's a 'Trigger Condition' field with the expression `trigger.getDay() > 0 && trigger.getDay() < 6`. The main area is divided into three columns:

- Left Column (Group):** Contains a 'Task' section with a 'StartShow' button and a 'Realm Variable' section with 'ProjectorPower' set to 'false'.
- Middle Column (Tasks):** A list of 7 tasks:
 - do Network.Barco1.power, set to ProjectorPower.value
 - do Network.Barco2.power, set to ProjectorPower.value
 - do Network.Barco3.power, set to ProjectorPower.value
 - do WATCHOUT.WO2.wakeUp, timeout: Literal value or expression
 - do WATCHOUT.WO2.load, showName: "Third Generation Intro"
 - do WATCHOUT.WO2.play, timeline: Literal value or expression
 - do IO.DALI_5.value, set to !ProjectorPower.value
- Right Column (TASK Configuration):**
 - StartShow button
 - Enabled checkbox (checked)
 - Trigger: Time of Day dropdown
 - Time: 9:00
 - Starting at Date checkbox (unchecked)
 - Ending after Date checkbox (unchecked)
 - Repeat: At Interval dropdown
 - Interval: 1:00
 - Ending at Time checkbox (checked)
 - End Time: 18:00

Страница Задачи (Task) в приложении Blocks.

4. СПОТЫ

В этой главе представлены подробные описания разных типов спотов. Вводная информация – в разделе «[Спот](#)» в главе «Основные понятия». Вы найдете все свои споты на странице [Display](#) в списке слева. Споты могут быть внутри группы спотов ([Spot Group](#)), и чтобы войти в группу, нужно щёлкнуть на значок стрелки.



В крайнем левом столбце списка спотов (Spot list) содержится перечень свойств спотов. Перетащите правый край столбца спотов (Spot), чтобы открыть более подробную информацию. При этом станут видны ещё две колонки:

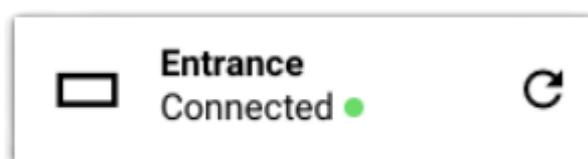
- **Назначенный блок (Assigned Block)** указывает тип и имя блока, который в данный момент назначен этому споту.
- **Воспроизводимый блок (Playing Block)** указывает текущий воспроизводимый блок (выделен серым цветом, если спот находится в автономном режиме). Это часто совпадает с назначенным блоком, но может и отличаться, если назначенный блок является Расписанием (Schedule) или если назначенный блок был переопределён блоком приоритета, что можно сделать с помощью кнопки или задачи.

[?] Если в символе назначенного или воспроизводимого блока отображается вопросительный знак, это означает, что блок не найден, и поэтому не может быть воспроизведён. Это происходит, если вы установили несуществующий блок с помощью кнопки или задачи или если вы удалили/переименовали блок после того, как назначили его споту.

Чтобы заставить спот (Spot) воспроизводить определённый блок, просто перетащите блок на спот на странице [Display](#), см. «[Отображение блока](#)». Если вы перетащите блок в группу ([Spot Group](#)), блок будет воспроизводиться на всех спотах этой группы, если только он не будет переопределен для какого-то из них.

Статус спота

Статус подключения спотов отображается списком. Если спот онлайн, отображается символ перезагрузки. А если расширить столбец Spot, отображается индикатор подключения (Connected) или для гостевого спота (Visitor) – количество текущих подключений.



После внесения изменений в блок щелкните символ перезагрузки, чтобы отобразить исправленный блок. Удерживая Shift при нажатии на символ перезагрузки, вы инициируете также перезагрузку программного обеспечения дисплея.

Настройки спота (Spot Settings)

Дважды щёлкните на значок, чтобы изменить настройки спота. Настройки зависят от типа спота, как правило, это имя, тэг и идентификатор местоположения (Name, Tag, Location ID).

Имя спота

Используется для распознавания спота в пользовательском интерфейсе, а также для управления им при помощи кнопок и задач.

▲ Избегайте изменения имени спота, если вы запрограммировали какие-либо действия (кнопками или задачами), связанные с этим спотом, так как это нарушает соединение.

Тэги (Tags)

Вы можете назначить любой набор тэгов споту. Тэг – это произвольное слово или комбинация слов. Тэги разделены запятой. Тэги могут быть использованы вместе с блоком выбора тэгов ([Tag Selector](#)) для выбора контента для этого спота. Дополнительные тэги могут быть применены к споту через кнопку [Button](#) (используется в основном для гостевых спотов Visitor).

ID местоположения (Location ID)

Споту присваивается короткий числовой идентификатор (ID). Поиск спота происходит с помощью блока [Locator](#) введением ID спота на клавиатуре. Это можно использовать для создания мобильного гида, когда посетители воспроизводят релевантный контент, вводя номер экспоната, показанный на этикетке. Вам нужно только назначить ID спотам, которые вы хотите использовать таким образом. Часто для этого

Дисплей-спот



Представляет один дисплей, обычно в фиксированном месте. Такие дисплеи/плееры должны подключаться к серверу Blocks по проводной сети. Помимо имени, тэгов и ID (см. выше), он имеет нижеперечисленные настройки.

▲ Серверы Blocks лицензированы для ограниченного количества дисплеев и мобильных спотов. Если вы попытаетесь добавить больше дисплей-спотов, чем это предусмотрено лицензией, сообщение будет выведено на экран, и в этом случае нужно связаться с PIXILAB, чтобы расширить свою лицензию.

Основные (General)

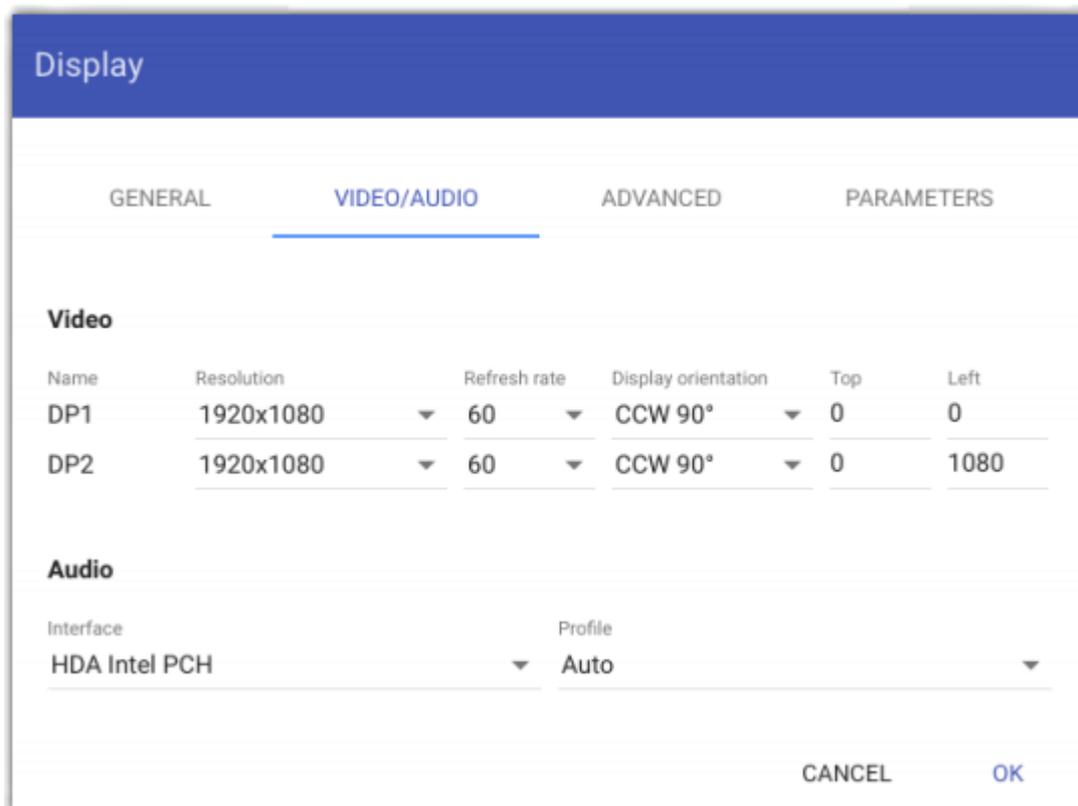
Основные настройки, используемые и для других типов спотов, как было описано выше.

Re-assign to Spot ID (переназначить на Spot ID)

Эта опция появляется только при подключении новых неназначенных дисплеев. Опция позволяет вам переместить дисплей-спот на один из неназначенных. Это бывает полезно в тех случаях, когда вы хотите заменить физический плеер другим плеером, сохраняя все настройки без изменений. В зависимости от типа и модели назначаемого плеера, некоторые настройки могут больше не применяться. Например, некоторые настройки из раздела «Видео / Аудио» могут больше не применяться, если у нового плеера нет той же аппаратной конфигурации, что и у старого.

Video/Audio

Поля, найденные под этим заголовком, зависят от типа плеера. Основные настройки включают в себя ширину и высоту дисплея, как указано при добавлении в Blocks. При использовании плеера с PIXILAB версии 3 и старше предоставляются следующие опции.



Name, Resolution, Refresh Rate, Display Orientation (имя, разрешение, частота обновления и ориентация экрана)

Управление разрешением, частотой обновления и ориентацией экрана для каждого дисплея, подключенного к дисплей-споту. Столбец «Name» отображается только для плееров с несколькими выходами, и в этом случае относительное положение каждого дисплея и другие настройки могут быть установлены независимо. Это позволяет вам расположить дисплеи по горизонтали, вертикали или в виде сетки. Тщательное позиционирование дисплеев позволяет компенсировать ширину рамок дисплеев или указывать перекрывающиеся области изображения для проекторов со встроенной функцией смешивания краев.

Обратите внимание, что поля «Name» могут не соответствовать физическим соединениям. Например, на иллюстрации выше выходы называются DP1 и DP2, хотя оба они являются выходами HDMI.

▲ В то время как несколько дисплеев могут быть подключены к одному плееру, плеер по-прежнему представляет собой один дисплей-спот. Его блок охватывает объединенное пиксельное пространство всех подключённых к плееру дисплеев.

Audio (настройки звука)

Выбирает устройство вывода звука и его конфигурацию. Это полезно, если доступно несколько вариантов вывода звука, например, аналоговый и HDMI, чтобы быть уверенными, что используется нужный выход.

Advanced

Некоторые редко используемые параметры, включая требования авторизации для дисплей-спота.

Small Blocks

Как и дисплеи, большинство типов блоков имеют определённые размеры. Если блок меньше дисплея, то параметр Small Blocks будет определять, что произойдет:

- **Scale to Fit (уместить на дисплее)** - масштабирует блок (сохраняя соотношение его сторон) для заполнения дисплея.
- **Center (по центру)** блок в середине дисплея, без масштабирования.
- **Top Left (в верхнем левом углу)** помещает блок в угол дисплея. Эта функция удобна при использовании с процессором светодиодной видеостены или аналогичным, и в этом случае предпочтительно всегда помещать блок в известное положение, например, в верхний левый угол.

Role Required to Use Spot (доступ к использованию спота)

Ограничивает доступ к использованию и управлению спотом. Это важно для планшетов персонала и обеспечивает контроль за функциями, которые не должны быть доступны для посторонних. Если установлено значение, отличное от «Нет», будет запрашиваться авторизация для предоставления доступа к споту. Подробнее об этом см. [«Авторизация и роли»](#).

Role Required to Control (доступ к контролю)

Доступ к управлению свойствами спота, такими как назначенный ему блок или состояние включения / выключения. Подробнее об этом см. [«Авторизация и роли»](#).

Power Down when no Block shown (отключение питания, когда блок не отображается)

Если выбрана эта опция, и у дисплей-спота есть возможность для управления питанием, оно будет отключено, когда споту не назначен блок.

Это особенно полезно в сочетании с расписанием ([Schedule](#)), так как оно будет автоматически включать / выключать электропитание, выключая питание всякий раз, когда на споте не отображается ни один блок.

Preferred Synchronizer Master in Group (мастер синхронизации в группе)

Эта опция существует только для дисплей-спота внутри [группы спотов](#). При выборе этой опции дисплей-спот отображается как мастер при воспроизведении блока синхронизации ([Synchronizer](#)) на спотах этой группы (например, путем перетаскивания блока в группу спотов). Если ничего не выбрано, Blocks выберет мастера автоматически, поэтому в большинстве случаев вам не нужно указывать это. Вы можете назначить мастера, если в какой-то момент придётся заимствовать дисплеи из этой группы для других целей. Определите спот, который не будет заимствован, в качестве мастера, чтобы оставшиеся споты работали должным образом.

Parameters (параметры)

Назначает начальные значения параметрам, связанным с этим спотом. См. «[Параметризация путей свойств](#)» для получения подробной информации о том, как использовать эту функцию.

Name	Initial Value	Comment
toPlay	DS1	Spot to play/stop from this panel



Visitor (гость, посетитель)

Представляет мобильные устройства, такие как смартфон, планшет iPad или Android. Гостевые споты могут быть такими:

- личными смартфонами посетителей (что предполагает наличие доступной для посетителей локальной сети Wi-Fi);
- прокатными устройствами для посетителей;
- управляющими планшетами операторов (для персонала).

Размеры гостевого спота

В отличие от обычных дисплей-спотов, когда вы знаете точные размеры дисплея при добавлении спота, гостевой спот (Visitor spot) может быть использован на разных устройствах с разными размерами экрана, разрешениями и ориентацией. Это особенно актуально, если вы разрешаете посетителям использовать их собственные мобильные устройства. Следовательно, при добавлении спота Visitor, как правило, приходится рассматривать наименьший общий знаменатель с точки зрения размера экрана.

Использование более крупного или более продвинутого дисплея обычно не является проблемой (за исключением того, что кнопки и другие элементы управления могут выглядеть слишком большими). Однако ориентация при проектировании на более продвинутый дисплей, чем у большинства ваших пользователей, может привести к тому, что некоторые элементы исчезнут с экранов пользователей. Поэтому лучше быть немного консервативными. Например, можно принять, что дисплей iPhone 5 является подходящим общим знаменателем. Разрешение экрана iPhone 5 составляет 1136 пикселей (высота) на 640 пикселей, но некоторые пиксели теряются в адресной строке вверху и кнопках навигации внизу, в результате получается около 920 на 640 пикселей эффективной области отображения.



Обратите внимание, что точные значения количества пикселей не так важны. Фактически вы определяете соотношение сторон и координатное пространство для размещения контента. Хотя всё становится проще, если размер контента и дисплея совпадают, контент в конечном итоге будет масштабироваться в соответствии с шириной любого мобильного устройства, на котором он будет представлен.

Доступ к гостевому споту

В отличие от дисплей-спотов (Display Spot), где есть взаимно однозначное соответствие между физическим дисплеем и элементом в списке спотов, любое количество физических устройств может быть связано с одним гостевым спотом из этого списка. Все мобильные устройства, подключённые к URL-адресу гостевого спота, будут отображать один и тот же контент. После добавления гостевого спота в список спотов можно получить доступ к нему с помощью URL-адреса, например:

<http://10.0.2.10:8080/spot/?mobile=Mob1>

Это очень похоже на URL-адрес, используемый для дисплей-спота (Display Spot), см. раздел «[Добавление дисплея](#)». Замените 10.0.2.10 на IP-адрес вашего сервера. Число «:8080» после IP-адреса – это номером порта, используемого сервером. Этот номер может быть другим в зависимости от конфигурации вашего сервера. Если вы используете стандартный http-порт «:80», то не нужно указывать эту часть в URL-адресе.

▲ Причина, по которой сервер Blocks предварительно сконфигурирован для работы через порт 8080 вместо стандартного 80, заключается в том, что некоторые операционные системы не позволяют приложениям использовать порты с номером меньшим, чем 1024 без авторизации. Windows не имеет этого ограничения, что упрощает переход на порт 80, если это необходимо. Но для согласованности по умолчанию и под Windows порт имеет номер 8080. Для нашего сервера Linux может быть использован стандартный порт 80, так как предусмотрена внутренняя переадресация.

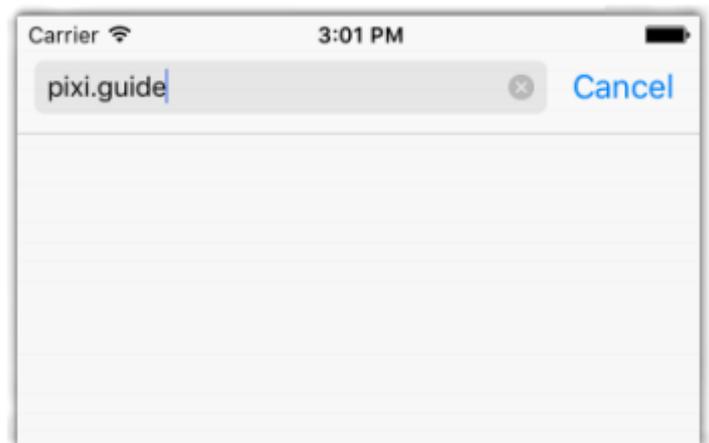
Последняя часть адреса, начиная с вопросительного знака, указывает, что это гостевой спот (Visitor). Имя, введенное после знака равенства, является именем гостевого спота, здесь «Mob1». Если Visitor Spot находится внутри группы спотов ([Spot Group](#)), то нужно написать имя спота после имени группы (групп). Например, если спот находится внутри группы под названием “Mobiles”, вы должны использовать следующий URL-адрес:

<http://192.168.0.33:8080/spot/?mobile=Mobiles/Mob1>

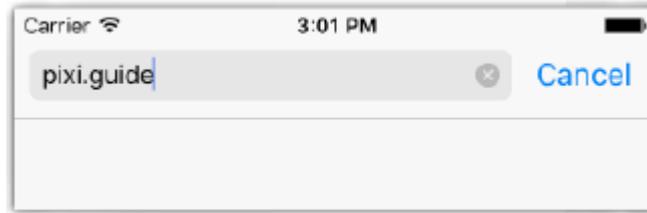
Группы спотов (Spot Groups) разделены косой чертой (слэшем), как показано выше. Хотя такая конфигурация подходит для локально управляемых устройств и устройств, выдаваемых напрокат, это не очень удобно для посетителей. Чтобы упростить регистрацию посетителей, вам необходимо изменить некоторые параметры сервера:

- Установите порт сервера на 80.
- Установите на сервере перенаправление по умолчанию на «/ spot /? Mobile = Mob1» (или другой путь к нужному гостевому споту).

Оба этих параметра описаны в разделе «[Конфигурация сервера](#)» ([Server Configuration Options](#)) и на Blocks wiki. Как только это будет сделано, достаточно ввести только IP-адрес сервера в браузер. Чтобы ещё больше упростить процесс, вы можете развернуть внутренний DNS-сервер, который связывает легко запоминающееся имя с IP-адресом сервера. Этот DNS-сервер затем используется для



гостевой сети Wi-Fi. Предполагая, что вы указали имя «pixi.guide» для IP-адреса сервера (10.0.2.10 в приведенном выше примере), вы можете просто набрать pixi.guide в адресной строке браузера вашего мобильного телефона, чтобы получить доступ к функциям гида, см. рис. ниже.



▲ Версия [Linux](#) на сервере Blocks включает всё, что нужно, чтобы выполнить настройки, описанные выше, включая функционал DNS-сервера.

Location (местоположение)

Представляет просто некое физическое местоположение или объект интереса (например, автомобиль на иллюстрации в разделе «[Споты](#)» в главе «Основные понятия»). Это будет полезно в тех случаях, когда вы захотите связать контент с этим местоположением, чтобы он воспроизводился через мобильные устройства посетителей. Это может быть звуковой клип для аудиогuida, видео, текстовое описание или что-то еще, что вы хотите представить на этом месте.

Добавьте Location в список спотов (можно сначала создать [группу спотов Spot Group](#) для этого). Присвойте ему имя и ID. Этот ID должен быть передан посетителю через какой-либо вынужденный знак в этом месте. Свяжите контент, который вы хотите отобразить на мобильном устройстве посетителя, с этим спотом. С помощью блока [Locator](#) посетитель вводит ID спота для доступа к этому контенту.

WATCHOUT

 Спот WATCHOUT позволяет воспроизводить звук, синхронизированный с презентацией Dataton WATCHOUT, что иногда используется в кинотеатрах или других специальных местах, таких как проекция на изогнутый экран или какой-либо физический объект (мэппинг). WATCHOUT может воспроизводить звуковую дорожку, связанную с шоу, можно также предоставить описание на дополнительных языках через мобильные устройства посетителей.

▲ На экране WATCHOUT *не* появляется блок, связанный с WATCHOUT-спотом (это только звук!). Блок появляется на гостевых спотах (Visitor Spots) при навигации к местоположению WATCHOUT Spot.

Кроме добавления WATCHOUT spot следует добавить и настроить кластер WATCHOUT на странице Manage (управление). Вы также должны получить звуковую дорожку, в точности соответствующую временной шкале шоу WATCHOUT. Звуковая дорожка должна совпадать с временной шкалой с самого её начала, даже если контент на временной шкале не начинается с начала. Например, если контент на временной шкале WATCHOUT начинается через пять секунд, вы должны добавить эти пять секунд молчания в начале звукового клипа для точного соответствия шкал.

 A screenshot of the 'WATCHOUT Spot' configuration form. The form has a blue header with the title 'WATCHOUT Spot'. Below the header are several input fields: 'Name' with the value 'WO-Main', 'Tags' (empty), 'Location ID#' with the value '44', and 'WATCHOUT Cluster *' with the value 'WO'. There is a 'Timeline Name (empty for Main)' field which is currently empty. At the bottom, there is a checkbox labeled 'Start when Accessed by Visitor' which is checked. At the very bottom right, there are 'CANCEL' and 'OK' buttons.

▲ Чтобы звуковой файл воспроизводился на мобильных устройствах, он должен быть закодирован с использованием формата AAC, обычно хранящегося в файлах .m4a.

WATCHOUT Cluster (кластер WATCHOUT)

Введите сюда имя кластера, как указано на странице Manage (управление).

Timeline Name (имя временной шкалы)

Введите имя временной шкалы WATCHOUT для синхронизации. Оставьте это поле пустым для синхронизации с основной временной шкалой.

Start when Accessed by Visitor (запуск при появлении посетителя)

Если этот флажок установлен, приложение Blocks сообщает программе WATCHOUT, что нужно начать воспроизведение, когда посетитель обращается к споту, введя идентификатор местоположения (Location ID) этого спота в мобильное устройство. Используйте этот метод, чтобы начать просмотр WATCHOUT «по требованию» первым посетителем. Последующие посетители смогут присоединиться к просмотру в текущий момент времени и не будут перезапускать шоу.

Если шоу WATCHOUT запускается другим способом (например, по запланированной задаче [Task](#)), оставьте это поле без «галочки». В этом случае просмотр *не может* быть запущен посетителем, который обращается к его идентификатору местоположения (Location ID).

Группа спотов

Группа спотов (Spot Group) действует как «папка», позволяя группировать споты способами, которые имеют смысл для вашей инсталляции. Вы можете использовать группу только для организационных целей, представляя этаж или комнату в здании. Группы спотов могут быть вложены друг в друга, что позволяет иметь внешние группы для этажей с вложенными в них группами для комнат на каждом этаже.

Можно использовать группу спотов (Spot Group) для соединения родственных дисплеев. Например, у вас может быть набор из трёх дисплеев, работающих вместе. Поместите их в группу спотов (Spot Group), а затем назначьте контент всем трём дисплеям за один раз, перетаскив блок в группу спотов (Spot Group), не тратя времени на каждый отдельный спот. Используя тэги для спотов внутри группы в сочетании с блоком выбора тэгов (Tag Selector), вы можете настроить распределение контента между дисплеями, например, пометить дисплеи слева, в центре и справа, и соответственно настроить селектор тэгов ([Tag Selector](#)).

Наконец, вы можете использовать группу спотов для синхронизации воспроизведения видео на нескольких дисплей-спотах:

- Поместите эти дисплей-споты в группу спотов.
- Отметьте каждый дисплей, если необходимо (например *left*, *center* и *right*).
- Используйте Synchronizer для воспроизведения видео с тегами, назначенными для каждого из видео, чтобы ролики проигрывались на нужных дисплеях.
- Назначьте этот Synchronizer группе спотов (что позволит применить его ко всем спотам в группе, если они не отменены).
- Можно также назначить один из дисплеев в группе как [Synchronizer Master in Group](#) (мастер синхронизации в группе).

См. [«Синхронизация видео на нескольких дисплеях»](#).

5. БЛОКИ

Управление контентом осуществляется с использованием различных типов *блоков*. Хотя многие типы блоков могут показаться довольно простыми, их истинные широкие возможности становятся очевидными при использовании их в комбинации. Вводная информация – в разделе [Блоки](#) главы «Основные понятия».

Блок может быть либо на верхнем уровне страницы [Display](#), либо вложен в другой блок. Блоки верхнего уровня можно вручную назначить спотам (Spots), перетащив блок на спот (Spot) на странице Display. Они также могут быть динамически назначены при помощи кнопок и задач (buttons, tasks). Для автоматического изменения назначения блока в зависимости от времени, даты или дня недели используйте расписание ([Schedule](#)). Когда назначение блока изменяется, соответствующие дисплеи обновляются.

Создайте блок верхнего уровня, используя меню Block на странице Display, как показано справа. Многие типы блоков могут содержать внутри себя другие блоки. Например, слайд-шоу часто содержит изображения и видео, но также может содержать более сложные типы блоков, такие как Composition или Web.

Вот некоторые из наиболее часто используемых типов блоков:

- **Composition** (композиция) - для объединения нескольких меньших блоков, кнопок или других элементов управления на одной странице.
- **Book** (книга) - для размещения контента на нескольких страницах.
- **Slideshow** (слайд-шоу) - для показа последовательности изображений или других блоков с переходами.

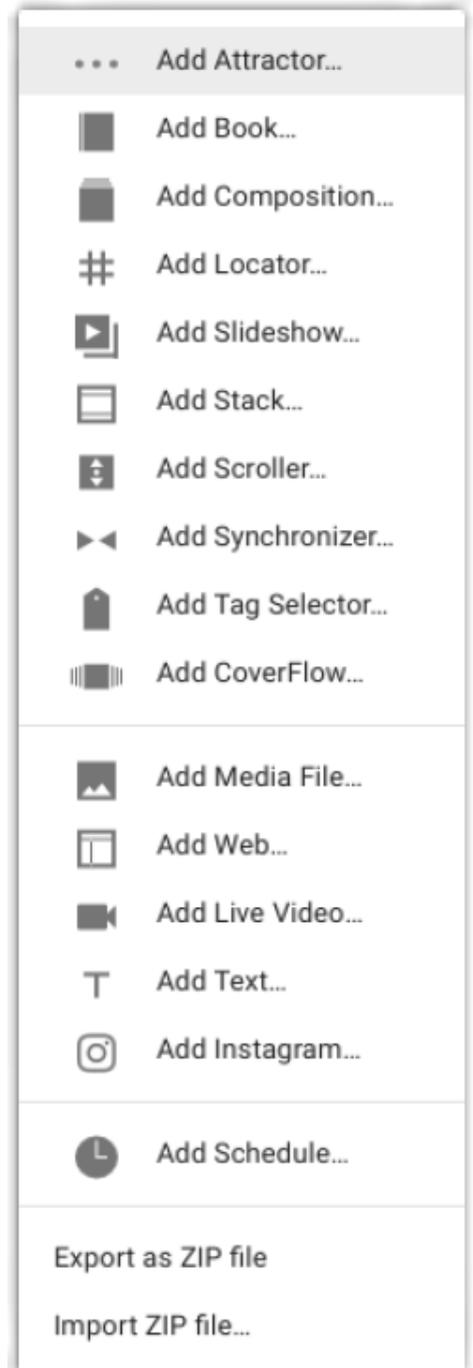
Группа блоков (Block Group)

При создании блока верхнего уровня вы присваиваете ему имя и связываете его с группой блоков. Изначально существует одна группа под названием Main (основная). Организация блоков по группам облегчает их поиск по мере увеличения количества блоков. Планируя заранее, заранее принимая решение, как назвать и сгруппировать блоки, вы сэкономите время при использовании этих блоков.

Когда у вас много блоков, иногда сложно найти блоки, с которыми вы работаете. Если блоки организованы в группы, вы можете отфильтровать набор блоков, отображаемых на странице Display, с помощью меню Group (Группа). Здесь вы можете выбрать просмотр только одной группы или блоков из всех групп.

Ограничение доступа к блокам с помощью ролей

Связывая группы и отдельные блоки с ролями, вы можете скрыть блоки для пользователей, у которых нет назначенной или более высокой роли. Ограничение доступа к блокам может сделать систему более удобной и менее запутанной. См. «[Авторизация и роли](#)» для получения подробной информации.



Чтобы ограничить видимость всех блоков в группе при помощи ролей:

1. Выберите нужную группу в меню Group на странице Display.
2. Нажмите символ замка рядом с названием группы.
3. Выберите необходимую роль в диалоговом окне.

Чтобы установить видимость отдельных блоков:

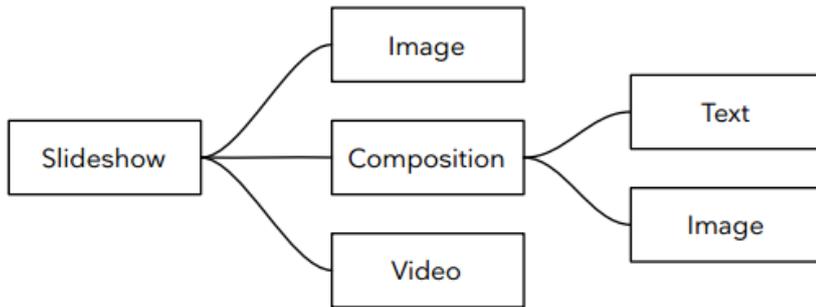
1. Дважды щелкните блок на странице Display, чтобы просмотреть его редактор.
 2. Выберите роль в раскрывающемся меню “Role Required to See” (роль, необходимая для просмотра).
- Если вы установите роль на уровне группы, эта роль наследуется всеми блоками в этой группе, кроме тех, которые явно не переопределяют этот параметр.

▲ Чтобы проверить эффект, вы должны войти в систему как пользователь с ролью ниже той, которая указана для видимого блока.

Редактирование блоков

Большинство типов блоков предоставляют доступ к специализированному редактору, который можно открыть, дважды щёлкнув на блок на странице Display. Щёлкните на логотип в левом верхнем углу, чтобы вернуться на страницу Display.

Блоки могут содержать другие блоки в древовидной структуре. Например, у вас может быть иерархия блоков, которая выглядит так:



Блок, содержащий вложенные блоки, например, блок Composition, как показано на рисунке выше, можно открыть, дважды щёлкнув на него в списке дочерних блоков ([Child Blocks](#)). Внутри такого вложенного блока в нижней части окна отображается строка навигации, путь к этому блоку. Щёлкнув на имя внешнего блока, вы вернётесь на уровень этого блока.



Строка навигации показывает типы и имена вмещающих блоков, что позволяет вам вернуться к любому внешнему уровню.

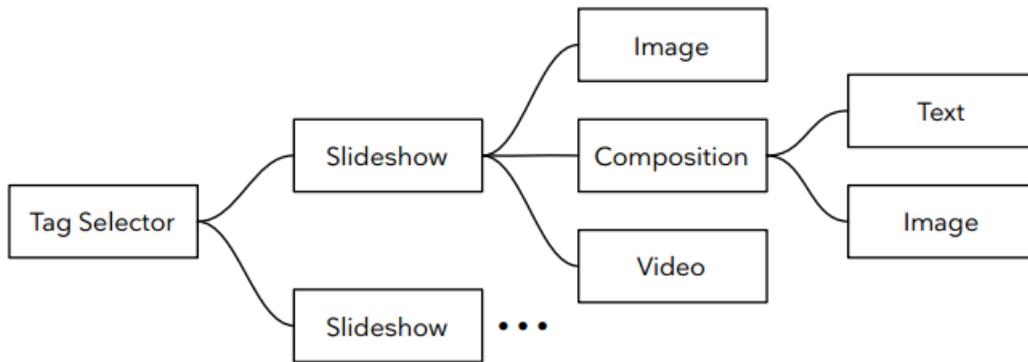
Добавление дочерних блоков (Child Blocks)

Добавьте дочерний блок, перетащив мультимедийные файлы в список дочерних блоков ([Child Blocks](#) list) или через меню “Add Child” (добавить дочерний блок). Различные типы блоков описаны далее в этой главе. Про изображение, видео и звук – см. раздел “[Media File](#)” в этой главе.

Сборка корневого блока (Wrapping a Root Block)

Меню «Wrap With» отображается только на самом верхнем уровне блока и позволяет вставить новый блок на корневом уровне, перемещая предыдущий уровень, бывший корневым, на уровень ниже, внутрь добавляемого блока. Например, начав с иерархии блоков со слайд-шоу в качестве внешнего блока (см. рис. выше), вы можете через некоторое время понять, что вам нужен селектор тэгов ([Tag Selector](#)) в качестве внешнего блока, чтобы представить два разных слайд-шоу в зависимости от некоторого тэга (например,

для поддержки нескольких языков). Для этого откройте редактор слайд-шоу (Slideshow editor), затем выберите Wrap With > Tag Selector, чтобы вставить селектор тэгов в качестве корневого блока, перемещая слайд-шоу на один уровень вниз:



Сборка корневого блока слайд-шоу при помощи селектора тэгов. Это позволяет добавлять другие блоки, например второе альтернативное слайд-шоу.

Экспорт и импорт блоков

Для экспорта и импорта блоков, а также для архивации блоков или передачи их в другую систему используйте команды в нижней части меню Block. Чтобы экспортировать блок, выберите его, затем скомандуйте “Export as ZIP file”. При этом на ваш компьютер загружается ZIP-файл с названием блока. Такой ZIP-файл впоследствии можно импортировать с помощью команды “Import ZIP file”. Выберите ZIP-файл, содержащий ранее экспортированный блок, затем укажите группу и имя для импорта блока.

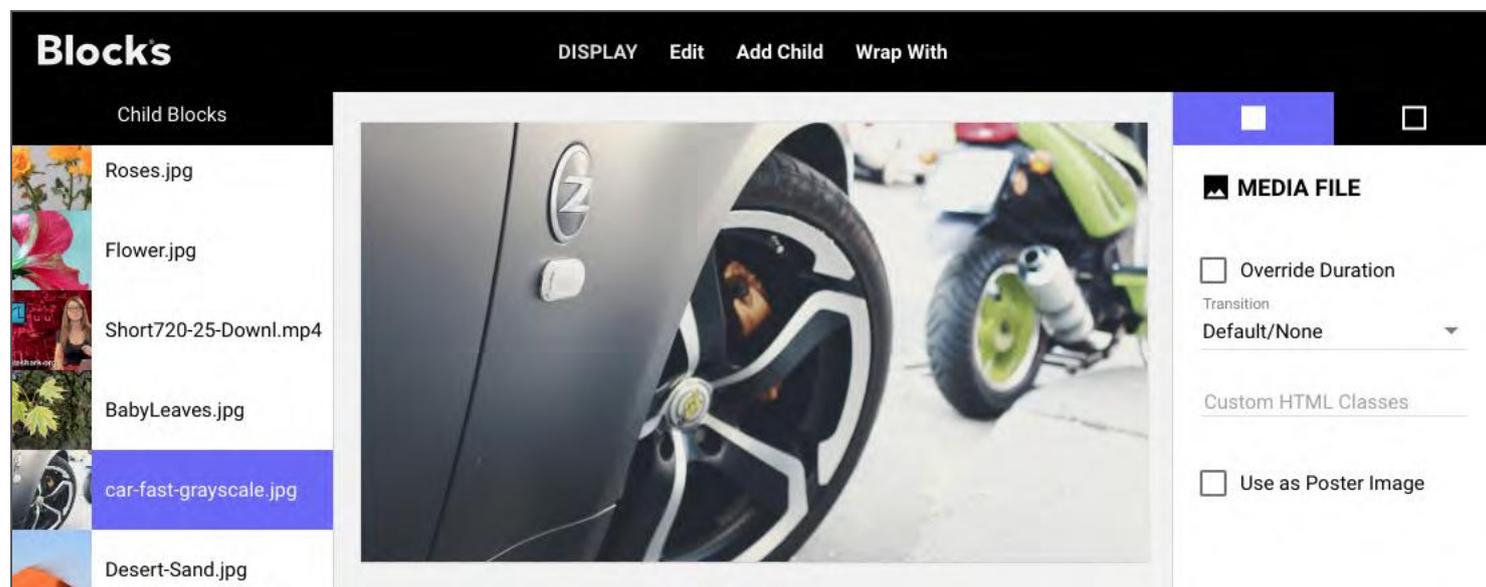
Использование блоков в качестве шаблонов

Если вы часто создаете похожие блоки, вы можете сэкономить время, сначала создав блок с базовым контентом. Затем продублируйте этот блок в качестве отправной точки для каждого отдельного блока. Сохраните эти шаблоны в их группах, например, с именем Templates (Шаблоны), затем используйте команду Duplicate (дублировать) из меню Edit, чтобы продублировать шаблон, присвоив ему новое имя и поместив вновь образованный файл в соответствующую группу.

Слайд-шоу



Слайд-шоу представляет свои дочерние блоки последовательно, с переходом между ними.



Child Blocks (дочерние блоки)

Дочерние блоки, из которых составляют слайд-шоу, отображены списком в левой части окна редактора. Изменить порядок появления блоков можно простым перетаскиванием. Дважды щёлкните на составной блок, чтобы открыть его редактор (не относится к простым блокам, таким как изображения).

Щёлкните дочерний блок в списке или сплошной квадрат в верхнем правом углу, чтобы просмотреть настройки выбранного дочернего блока. Эти настройки различны для блоков разных типов и зависят также от типа внешнего блока.

Override Duration (переопределить длительность)

По умолчанию слайд отображается в течение времени, указанного в настройках слайд-шоу (см. далее). Видео отображается в течение всей его продолжительности. Щёлкните на **Override Duration**, чтобы изменить продолжительность конкретного дочернего блока, затем установите нужную продолжительность, перетащив ползунок или введя числовое значение в это поле. Ползунком можно задавать продолжительность только до 10 секунд, в поле вы можете ввести любую продолжительность.

Transition (переход)

Если установлено значение «По умолчанию/Нет» (Default/None), то будет использован переход, указанный для слайд-шоу в настройках его внешнего блока (см. ниже). Выберите другой тип, и этот переход будет действовать только для этого блока.

Custom HTML Classes (пользовательские классы HTML)

Применяет к этому блоку класс HTML (или несколько классов, разделенных пробелами), позволяя конструировать блок особым образом в сочетании с пользовательскими CSS. Дополнительную информацию см. [«Пользовательские стили» \(Custom Styling\)](#).

Use as Poster Image (использовать как заставку)

Выбирает дочерний блок для представления блока на странице Display. По умолчанию это первый видимый дочерний блок. Вы можете выбрать более представительный дочерний блок, используя этот флажок. Этот параметр не влияет на воспроизведение блока.

Настройки внешнего блока

Щёлкните на пустой квадратик в правом верхнем углу, чтобы отредактировать настройки слайд-шоу в целом. Щёлкните на символ пера рядом с названием блока, чтобы переименовать этот блок или переместить его в другую [группу блоков](#).

▲ Переименование или перемещение блока, который используется, приведет к нарушению ссылок на этот блок (например, при назначении спота), и вам придётся восстанавливать их вручную.

Width/Height (ширина/высота)

Размеры блока. Вы устанавливаете эти размеры при первоначальном создании блока. Здесь можно изменить его настройки. Как правило, вы не изменяете эти настройки после создания блока. Если такое случится, следует пересмотреть места, где используется этот блок, чтобы убедиться, что он выглядит так, как ожидалось. Как правило, размеры блока должны соответствовать разрешению спота, на котором он будет отображаться, или, по крайней мере, иметь такое же соотношение сторон.

Custom CSS URL and Custom HTML Classes (пользовательские URL-адрес CSS и классы HTML)

Эти поля позволяют настраивать стиль кнопок и других элементов. “Custom CSS URL” (пользовательский URL CSS) применяется только к блокам верхнего уровня. Это продвинутая функция. Подробнее – в материале [«Пользовательские стили» \(Custom Styling\)](#).

Play Automatically, Loop (автовоспроизведение и цикл)

В большинстве случаев требуется, чтобы слайд-шоу воспроизводилось сразу. Если слайд-шоу является блоком верхнего уровня, вы, как правило, хотите его зациклить. Если слайд-шоу - внутренний блок (например, во внешнем слайд-шоу), вы можете захотеть, чтобы внутреннее слайд-шоу воспроизводилось только один раз, а затем внешний блок продолжил свою работу.

Pause Slides when Slideshow Paused (пауза, когда слайд-шоу на паузе)

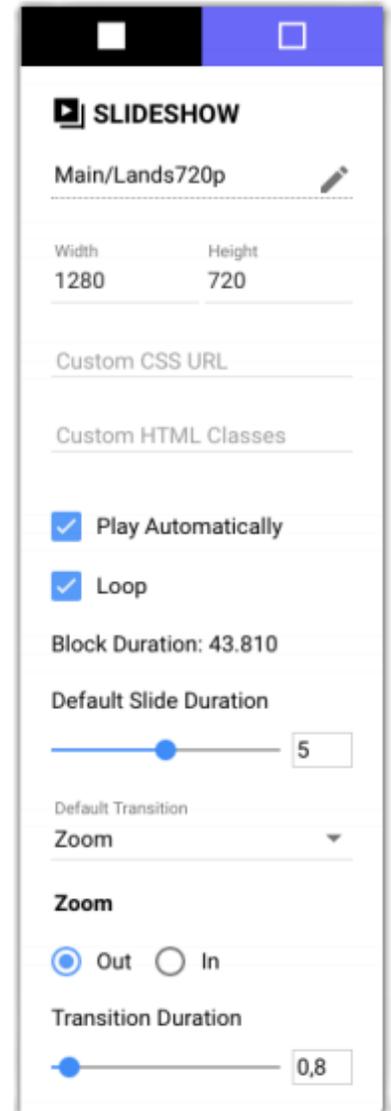
Если выбрано, текущий слайд будет на паузе, когда показ слайдов будет приостановлен. Это бывает полезно для слайдов, содержащих видео. Если этот параметр не выбран, слайд-шоу будет приостановлено, но любое видео продолжит работу до конца и остановится, ожидая, пока слайд-шоу не возобновит воспроизведение.

Default Slide Duration (продолжительность слайда по умолчанию)

Определяет, как долго показывать дочерние блоки, которые не имеют заданной продолжительности, например неподвижные изображения. Видео или другой дочерний блок, который имеет заданную продолжительность, будет показан полностью (если вы не переопределите продолжительность показа этого дочернего блока).

Default Transition (переход по умолчанию)

Переход, используемый между дочерними блоками в этом слайд-шоу, если не переопределено для дочернего блока. Доступны следующие типы переходов:

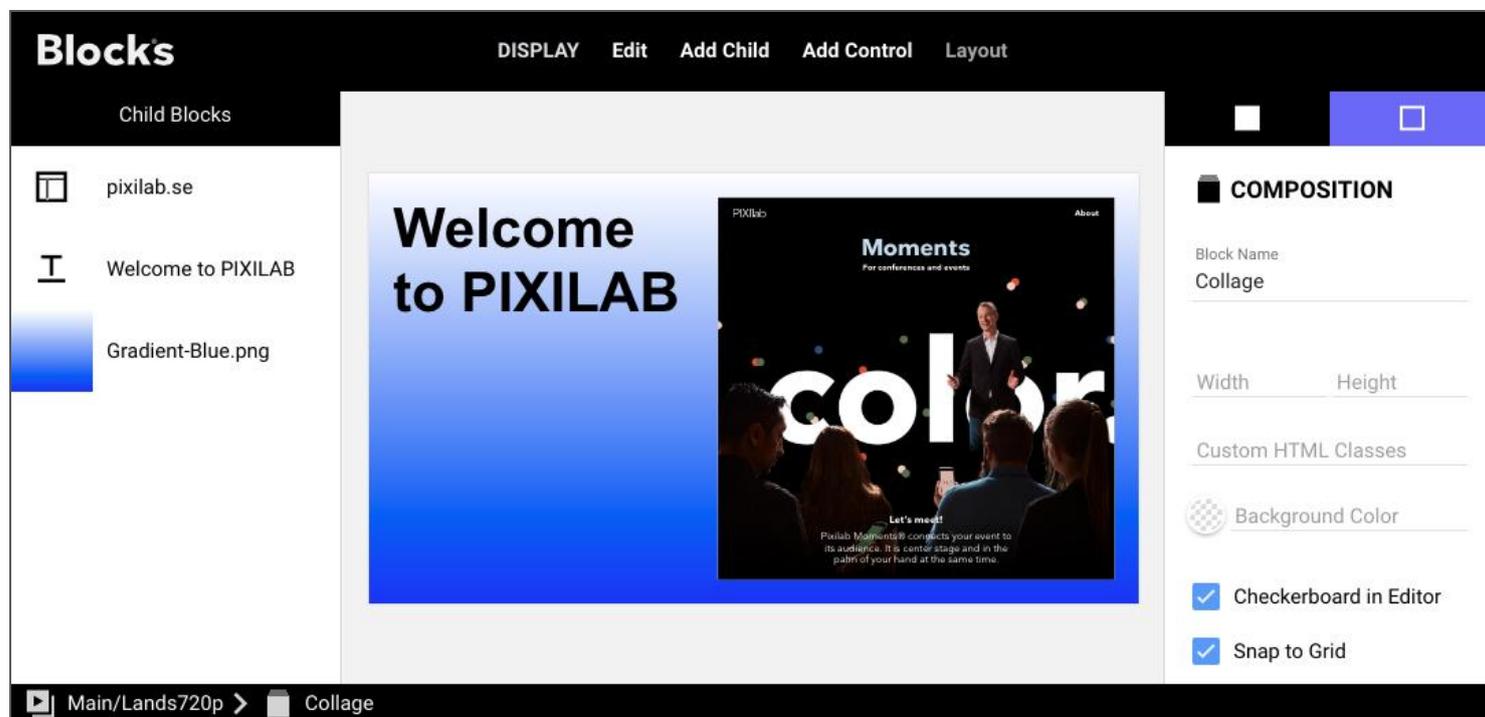


- **Cut (прямая склейка).** Не происходит никаких специальных эффектов при переходе. Новое изображение появляется мгновенно.
- **Dissolve (растворение).** Плавный переход в новое изображение.
- **Slide In (слайд).** Новое изображение скользит поверх старого. Можно указать направление соскальзывания.
- **Push (вытеснение).** Новое изображение наползает по мере того, как старое изображение «отползает».
- **Pop (всплывание).** Новое изображение всплывает поверх старого из-за него.
- **Zoom (масштаб).** Старое изображение исчезает и масштабируется вниз, когда появляется новое, дополнительно - с эффектом «вращения».

Для большинства переходов возможно указать продолжительность (то есть его скорость). Некоторые переходы имеют дополнительные настройки. Чтобы увидеть действие этих настроек, измените настройки, щёлкните на предыдущий слайд в списке, а потом вернитесь к текущему слайду.

Composition (композиция)

Блок Composition отображает свои дочерние блоки вместе, рядом или друг над другом, или даже в обратном порядке. На рисунке, показанном ниже, синяя градиентная заливка является самым нижним слоем, а блоки Текст (Text) и Веб-сайт (Website) расположены сверху. Перетащите дочерние блоки в списке, чтобы изменить порядок их укладки. Строка навигации, показанная внизу, показывает, что это вложенный блок. Он находится в слайд-шоу Lans720p, который является основным блоком группы.



Block Name (имя блока)

Дайте блоку осмысленное имя, чтобы его было легче найти внутри внешнего блока. Это имя также можно использовать при переходе к дочернему блоку внутри блоков Book, Slideshow или подобных. Избегайте изменения имени блока, если вы используете это имя, чтобы найти блок через кнопку, задачу или подобным образом.

Width/Height (ширина/высота)

Хотя иногда бывает полезно переопределить ширину и/или высоту внутреннего блока, в большинстве случаев вы оставляете эти поля пустыми, чтобы наследовать размеры внешнего блока. Если композиция – блок верхнего уровня, эти поля являются обязательными и определяют общий размер блока.

Background Color (цвет фона)

По умолчанию фон композиции прозрачен. Это позволяет использовать композиции во внешних композициях в качестве средства группировки внутренних блоков. Если вы хотите, чтобы в композиции был непрозрачный фон, выберите его цвет здесь. В приведенном выше примере это не имеет значения, так как синий градиент заполняет всю композицию.

Checkerboard in Editor («шахматная доска» в редакторе)

Определяет фон, используемый при редактировании композиции. Не применяется, если выбран цвет фона. Выберите стиль шахматной доски, это сделает ваш контент хорошо видимым при редактировании.

Editor Scale (масштаб для редактора)

Большие композиции обычно масштабируются автоматически, чтобы поместиться в редакторе. В некоторых случаях, например при редактировании текста, это может быть неудобно, так как текст будет слишком мелким. Поэтому выберите нужный масштабный коэффициент и прокрутите страницу, чтобы просмотреть нужную часть для редактирования.

Snap to Grid (привязать к сетке)

Если этот флажок установлен, элементы, перемещённые с помощью мыши, будут привязываться к каждому 20 пикселям от верхнего/левого угла. Это часто упрощает процесс перемещения. Снимите этот флажок, чтобы перемещать элементы свободно. Вы всегда можете позиционировать элементы где угодно числами или с помощью команд в меню Layout (макет), даже если этот параметр выбран.

Keep Children Inside (дочерние блоки внутри)

Если этот флажок установлен, дочерние блоки не могут быть передвинуты за пределы вмещающей композиции. Отмените этот выбор, если вы хотите использовать композицию, чтобы «обрезать» её внутренние блоки, показывая только ту их часть, которая пересекает композицию.

Show on Interaction (показать при воздействии)

Если этот флажок установлен, все дочерние блоки будут скрыты через несколько секунд бездействия. Коснитесь дисплея (или переместите мышь), чтобы они снова появились. Это полезно, когда интерактивные элементы управления, такие как кнопки, располагаются поверх контента, закрывая его. При выборе этой опции элементы управления будут скрыты автоматически. Просто коснитесь дисплея, чтобы они снова появились.

▲ Эта опция доступна только для вложенных композиций. Не для блоков композиции верхнего уровня.

Layout (макет)

В этом меню предоставлены команды для выравнивания выбранных блоков. Выберите нужные элементы либо в списке дочерних блоков (слева), либо с помощью мыши. Затем выберите команду для выравнивания выбранных блоков.

Align Horizontally (выровнять по горизонтали)

Перемещает выбранные блоки по горизонтали так, чтобы их центр, левая или правая стороны выстроились вдоль одной вертикали.

Align Vertically (выровнять по вертикали)

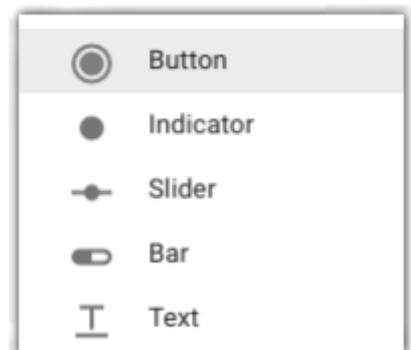
Перемещает выбранные блоки по вертикали так, чтобы их центр, левая или правая стороны выстроились вдоль одной горизонтали.

Set Width Ratio (задать соотношение по ширине)

Задайте ширину выбранного блока (-ов) в указанном соотношении размеров дочернего и вмещающего блоков. Это упрощает подбор размеров дочернего блока, чтобы заполнить половину (треть, четверть...) ширины вмещающего блока.

Add Control (добавить элемент управления)

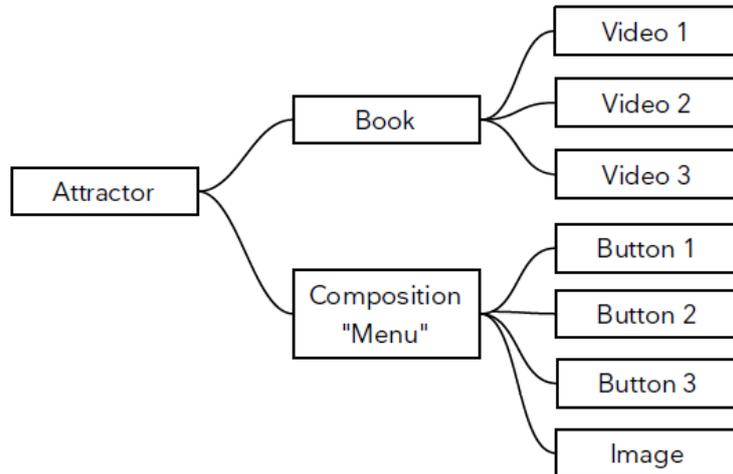
Элементы управления – это такие интерактивные элементы, как Кнопка (Button) или Ползунок (Slider), см. главу «[Панели управления](#)». Эти элементы считаются блоками, но они имеют смысл только в композиции, поскольку никогда не используются в полный размер экрана и должны некоторым образом располагаться друг по отношению к другу. Это также относится к текстовым блокам, которые добавляются через это же меню. Таким образом, в то время как меню Добавить дочерний блок (Add Child) доступно во всех блоках, которые могут иметь дочерние блоки, меню Добавить элемент управления (Add Control) доступно только при редактировании композиции.



▲ Если вы хотите использовать любой из этих типов блоков внутри другого блока, сначала добавьте композицию (Composition), а затем уже внутри этой композиции добавьте блок управления или текст.

Book (книга)

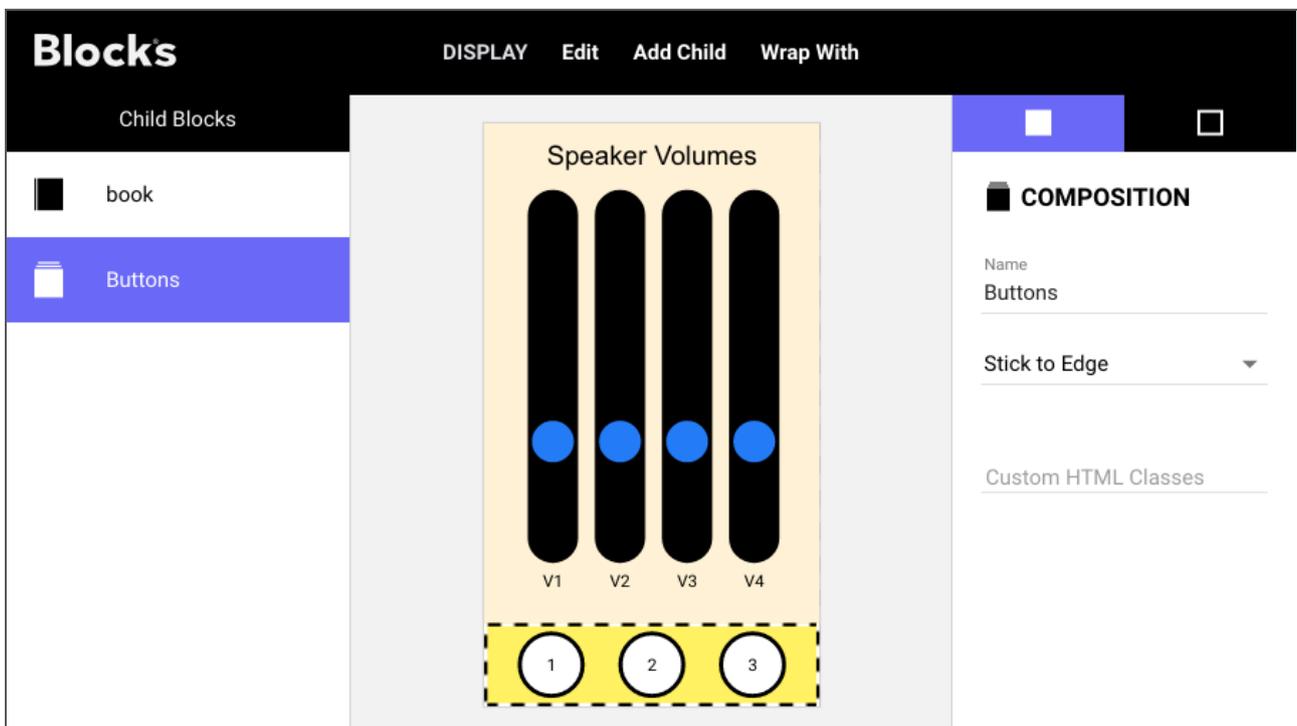
Книга отображает свои дочерние блоки как отдельные поименованные страницы. Вы можете перейти на страницу с помощью задач или кнопок, см. [«Поиск целевой страницы по имени» \(Locating a Target Page by Name\)](#) в разделе Button (кнопка) или [Task](#) (задача). В приведенном ниже примере показана книга с тремя страницами, каждая из которых демонстрирует видео. Кнопки в композиции «Menu» управляют книгой. Книгу и композицию «Меню» помещают в блок [Attractor](#), что заставляет меню каждый раз появляться, когда видео, показанное на странице в книге, заканчивается.



Stack (штабель)

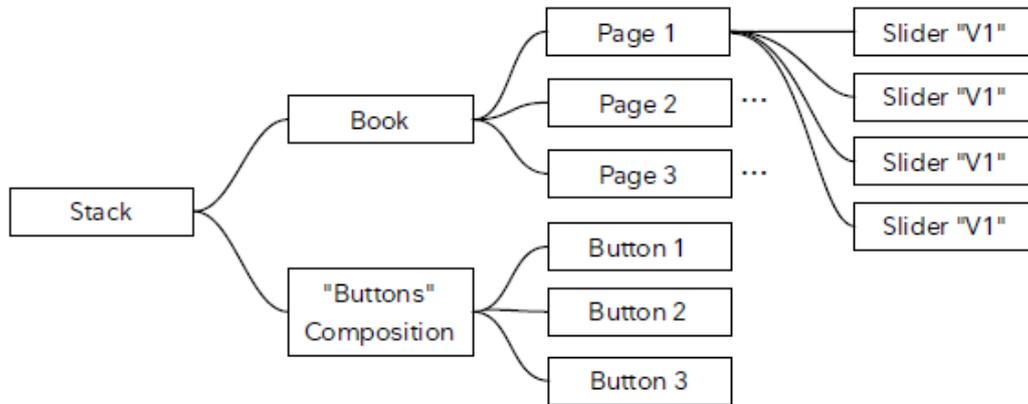


Стек отображает свои дочерние элементы в виде двух или более вертикально расположенных секций. Обычно он используется для добавления навигационных панелей в верхней и/или нижней части мобильного устройства (или аналогичного), при этом основная область занимает центр экрана.



Блок Stack с двумя дочерними блоками. Дочерний блок Book (четыре движка) занимает большую часть пространства. Композиция «Buttons» прикреплена к нижнему краю.

Дочерний блок может быть **Stick to Edge** (прикреплён к краю), как в случае с кнопками (Buttons) в примере, показанном выше. Такой дочерний блок «прилипает» к верхнему краю, если он помещен в верхней части списка, или к нижнему краю, если он расположен внизу. Блоку “Book” (это четыре ползунка в приведенном выше примере) затем назначают **Fill Remainder** (заполнить остаток).



▲ Можно создать аналогичную компоновку и используя композицию (Composition), однако только блок стек Stack гарантирует, что верхний/нижний блок с установкой «Прикрепить к краю» (Stick to Edge) всегда будет в верхней или нижней части экрана, независимо от размеров этого экрана, при этом основная область будет занимать оставшееся пространство. Это делает стек более адаптивным к разным размерам экрана мобильных устройств, в то время как дочерние блоки композиции занимают фиксированные положения.

Scroller (прокрутка)



Блок Scroller размещает свои дочерние элементы горизонтально или вертикально, позволяя зрителю вручную прокрутить результирующей макет в указанном направлении. Он также может использоваться с одним широким или высоким блоком, что позволяет прокручивать его. Дочерние блоки могут появляться вплотную рядом друг с другом или с установленным между ними промежутком.

▲ Поскольку этот блок явно предназначен для ручного взаимодействия, размещенные на нем видео не будут воспроизводиться автоматически. Более того, если вы отображаете несколько видео, запуск одного автоматически приостанавливает любое другое уже воспроизводимое видео.

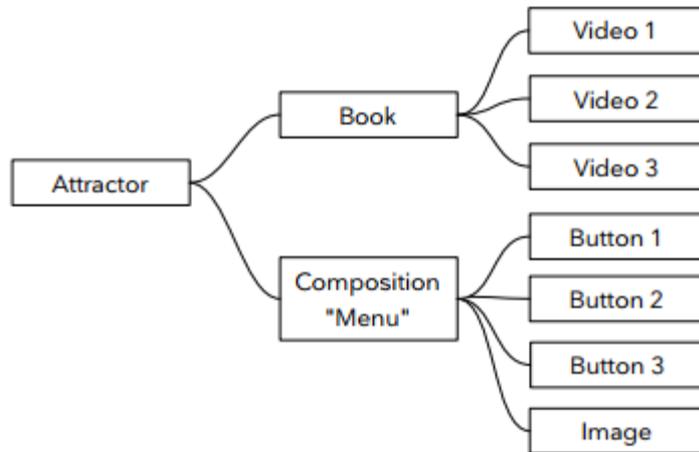
Размеры и масштабирование скроллера и его контента

Вы должны указать размер, перпендикулярный направлению прокрутки. Например, вы должны установить определённую высоту горизонтального скроллера.

Дочерние блоки скроллера, как правило, должны иметь размер, соответствующий указанной высоте или ширине. Сам Scroller будет масштабирован, чтобы соответствовать его ограничивающему блоку. В некоторых случаях, например при использовании в качестве основной области блока Stack, это масштабирование является динамическим и использует любое доступное пространство, которое будет зависеть от разрешения и ориентации устройства отображения.

Attractor •••

Блок Attractor имеет один *активный* дочерний блок, который всегда является его первым дочерним блоком, за которым следует один или несколько *пассивных* дочерних блоков. Это применяют для интерактивности, когда нужно показать что-то на экране, чтобы привлечь посетителей. Когда посетитель начинает взаимодействовать, дисплей переключается в активное состояние. Как только активное состояние заканчивается (например, когда видео перестает воспроизводиться), блок Attractor возвращается в пассивное состояние.



Блок Attractor с книгой Book в качестве активного дочернего блока и композицией Menu в качестве пассивного дочернего блока.

В приведенном выше примере показан блок Attractor в сочетании с другими блоками для создания интерактивного киоска. Здесь в пассивном состоянии отображается меню. Когда посетитель начинает взаимодействовать, нажав кнопку меню, чтобы начать воспроизводить видео, блок Attractor автоматически переключается в активное состояние, показывая одну из страниц книги (видео). Книга - первый, и следовательно, активный дочерний блок аттрактора. Как только видео заканчивается, блок Attractor автоматически возвращается в пассивное состояние, показывая композицию “Menu”.

Спот может быть активирован одним из следующих событий:

- При прикосновении. Применяется только в том случае, если выбрано “Activate by Touch/Click” (активировать касанием / щелчком).
- При использовании блока [Locator](#) для доступа к споту. Это позволяет посетителю активировать спот с помощью мобильного устройства.
- Программно, когда активное состояние спота установлено как true.

Спот и его Attractor возвращаются в пассивное состояние одним из следующих способов:

- При достижении конца воспроизведения (видео, слайд-шоу и т. д.). Применяется только если выбрано “Deactivate on Play End” (деактивировать при окончании воспроизведения).
- Через определенное время. Применяется только в том случае, если выбрано “Deactivate on Timeout” (деактивировать после перерыва) после указанного времени простоя.
- Программно, когда активное состояние спота установлено как false.

Использование нескольких пассивных дочерних блоков

Как правило, есть только один пассивный дочерний блок, как в приведенном выше примере, но таких блоков может быть сколько угодно. Если есть несколько пассивных дочерних блоков, Attractor будет чередовать их, и получатся некоторые вариации в цикле привлечения.

Tag Selector (селектор тэгов)

Блок Tag Selector работает в сочетании с тэгами, назначенными спотам. Тэги могут быть назначены статически (как описано в разделе “[Tags](#)” в главе «Споты»), или их можно назначать динамически и интерактивно при помощи кнопок. Типичное использование тэгов:

- Указание на то, что спот - это мобильное устройство, статически присваивая ему тэг «мобильный» (mobile).
- Указание на относительное положение спота в группе спотов (Spot Group); например «слева» (left), «в центре» (center) и «справа» (right), что обеспечивает подачу корректного контента на соответствующий дисплей в группе.
- Указание возрастной группы или языковых предпочтений посетителей. На мобильном устройстве может отображаться страница меню, такая как показана справа, что позволяет посетителям выбирать предпочтительный язык из набора поддерживаемых языков.



Svenska



English



Deutsch

Независимо от того, как тэг применяется к споту, тэг используется блоком выбора тэгов (Select Tag) при выборе подходящего дочернего блока из набора блоков для отображения на этом споте.

Блок Tag Selector отображает подходящий дочерний элемент. Здесь тэгом «ger» назначено воспроизведение немецкой версии видео.

Использование нескольких тэгов

Вы можете применить любое количество тэгов к споту. Вы можете также указать любое количество тэгов для дочерних блоков селектора тэгов (Tag Selector). При отображении селектора тэгов он покажет дочерний блок с наиболее подходящими тэгами. Если тэгов нет, он покажет первый дочерний блок.

Media File (медиафайл)

Блок этого типа представляет собой один мультимедийный файл. Добавьте его, либо выбрав «Media File ...» в меню Add Child, либо просто перетащив медиафайл в список дочерних блоков.

Можно использовать следующие типы файлов:

- Изображения форматов JPEG, PNG, SVG и GIF.
- Видео, кодированное кодеком H.264, как правило, в файл mp4 или совместимый.
- Звук кодируется кодеком AAC, как правило, в файл m4a или совместимый.

В общем, изображения и видео должны иметь те же размеры, что и блок, к которому вы их добавляете. Например, если размеры блока составляют 1920x1080, то изображения и видео должны иметь такие же размеры.

Исключением является блок композиция ([Composition](#)), где изображения и другие дочерние блоки, как правило, меньше самой композиции, чтобы их можно было разместить рядом или наложить друг на друга. Композиция позволяет масштабировать дочерние блоки, но все-таки лучше создавать контент с корректными размерами, это обеспечит лучшее качество и снизит нагрузку на воспроизводящее устройство, что особенно важно для видео и больших изображений.

Максимальное разрешение и битрейт для видео зависят от того, какое устройство вы планируете использовать в качестве дисплей-спота (Display Spot). Подробную информацию и рекомендации смотрите в документации производителя устройства воспроизведения.

Video Playback on Mobile (воспроизведение видео на мобильном)

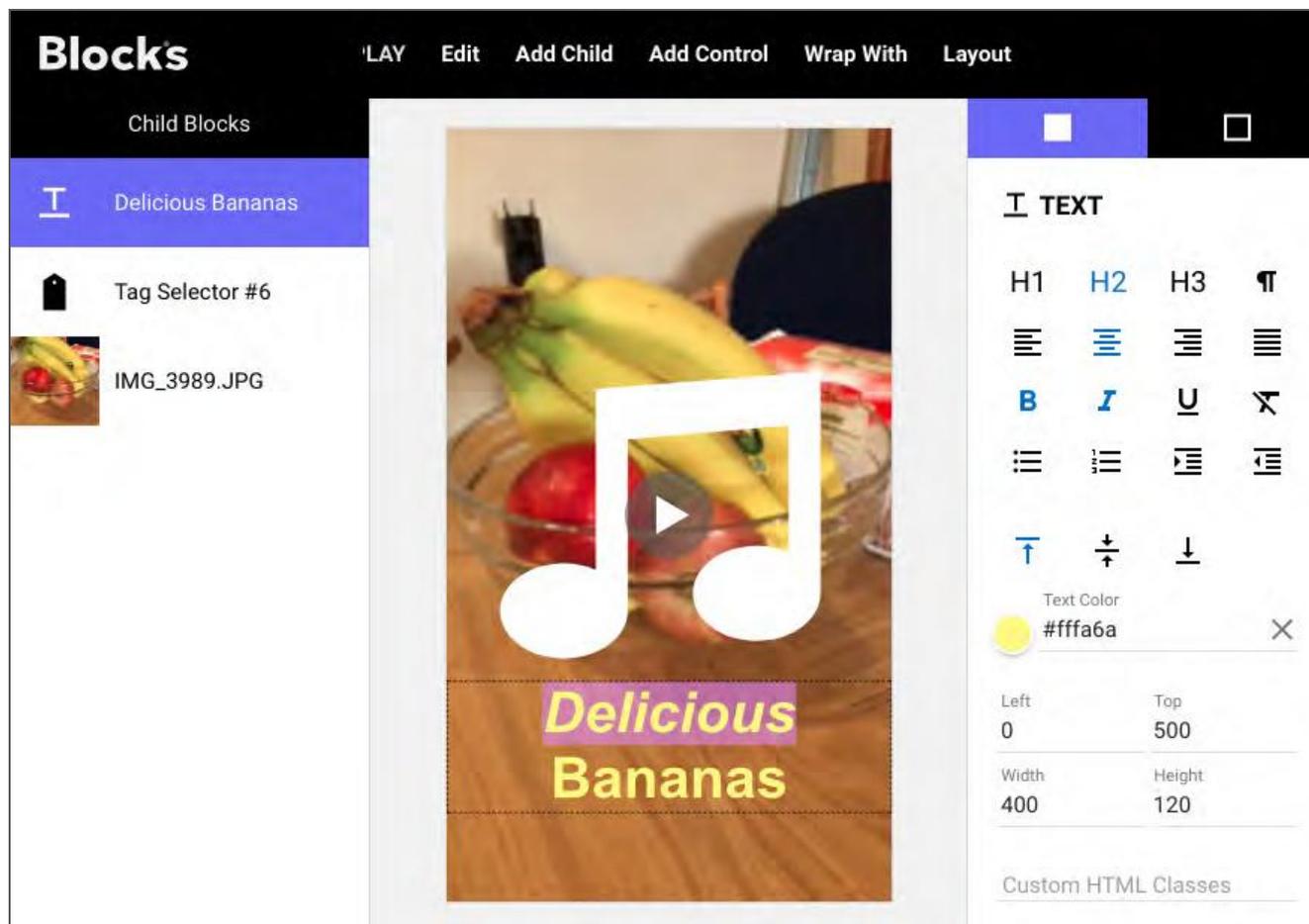
В некоторых контекстах видеофайлы имеют дополнительную настройку, контролирующую их поведение на некоторых мобильных устройствах (iPhone):

- **Fullscreen** (полноэкранный режим) автоматически увеличивает масштаб видео при воспроизведении. Это стандартный способ воспроизведения видео на iPhone.
- **Inline** заставляет видео оставаться в его текущем положении на экране и воспроизводить его там. Это позволяет другим смежным элементам оставаться видимыми, но также приводит к тому, что видео остается небольшим на экране, если пользователь не увеличил его до полного размера (нажав кнопку масштабирования, расположенную в углу рамки видео).

▲ Этот параметр может не повлиять на другие устройства отображения, кроме iPhone.

Text (текст)

Используйте текстовый блок для добавления текста в презентацию. Текстовый блок может быть только блоком верхнего уровня или внутри композиции. Если нужно использовать текстовый блок в другом месте, сначала добавьте композицию, затем добавьте текстовый блок внутрь композиции, выбрав “Text” в меню “Add Control”.



Текстовый блок с выделенным текстом «Delicious» (очень вкусный).

Внутри композиции (Composition) расположение текстового блока и его размеры можно задать мышью или числовыми значениями. Чтобы отредактировать текст, выделите его двойным щелчком мыши. Чтобы изменить размер и положение, а также другие атрибуты текста, используйте кнопки форматирования, расположенные с правой стороны. Кнопки H1, H2, H3 изменяют размер текста до трех заданных размеров заголовка. Кнопка “paragraph” возвращает текст до нормального размера. Второй ряд кнопок управляет выравниванием выбранного текста по горизонтали. Третья строка применяет варианты, такие как полужирный или курсив (с последней кнопкой, удаляющей любые такие переопределения формата).

Четвертый ряд кнопок можно использовать для добавления буллитов или цифр к абзацам, что полезно при отображении списков, и применять отступы. Последняя строка управляет выравниванием текста по вертикали внутри текстового блока. Здесь имеется только одна гарнитура шрифта, дополнительные шрифты могут быть добавлены и выбраны путем применения «Пользовательского класса HTML» (Custom HTML Class) к текстовому блоку (или охватывающему его блоку) в сочетании с пользовательскими стилями ([Custom Styling](#)).

Dynamic Text (динамический текст)

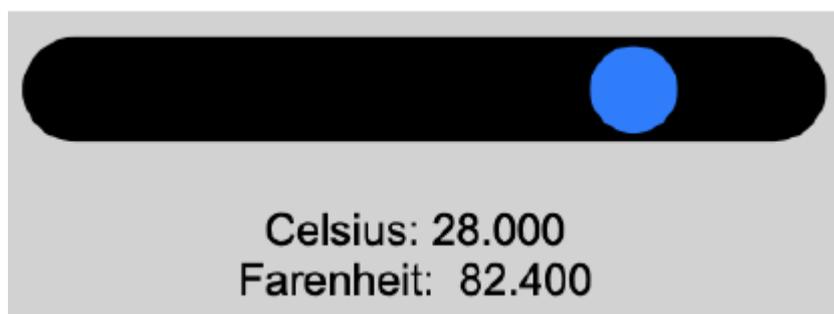
В дополнение к статическому тексту, описанному выше, блок Text также может отображать динамический текст, происходящий из некоторых свойств. Это похоже на работу индикаторов (Indicators and Bars), но показывает значение в виде текста, а не графически. После добавления свойства в текстовый блок, на свойство можно ссылаться внутри текста, используя специальный синтаксис, состоящий из пары фигурных скобок, содержащих знак доллара и однозначного номера индекса свойства (1-based property index number). Текст, показанный ниже, включает в себя два таких динамических свойства.



Первое свойство связано со значением переменной области с именем Centigrade, вставленной в текст как {\$ 1}. Аналогично, вторая привязана к другой переменной области с именем Fahrenheit и вставляется как {\$ 2}. Ползунок над текстом привязан к той же переменной области Цельсия. Задача запускается при изменении переменной Цельсия, устанавливая переменную Фаренгейта на основе переменной Цельсия, например:

```
1 set realm variable Fahrenheit .value to (Centigrade.value * 9/5) + 32
```

Отображая на устройстве с сенсорным экраном блок, содержащий вышеупомянутое, вы можете изменять значения с помощью ползунка, показывая температуру в градусах по Цельсию и Фаренгейту.



Форматирование динамического текста

В дополнение к порядковому номеру свойства некоторые типы данных также допускают дополнительные параметры форматирования. Эти параметры отделены двоеточием от индекса свойства (а также друг от друга, если используется более одного). Каждый форматор состоит из имени, знака равенства и значения. Применимые средства форматирования зависят от типа представляемого значения.

Name	Applies to	Value
true	boolean	Text to show for <i>true</i> value.
false	boolean	Text to show for <i>false</i> value.
scale	number	Factor to multiply by.
fixed	number	Number of decimal digits to show
show	time	Combination of <i>hmsf</i> characters, indicating which time components to include

Логическое значение по умолчанию отображается как *true* или *false*. Если вы хотите, чтобы он отображался как ДА или НЕТ (*YES* or *NO*), укажите свойство так:

`{$1:true=YES:false=NO}`

Число может быть отформатировано с использованием масштабного коэффициента (например, чтобы показать значение 0... 1 в виде процентов 0... 100) или с ограничением числа десятичных знаков. В примере, показанном выше, можно убрать десятичные знаки в значениях температур таким образом:

The image shows a control interface with a slider and a text display. The slider is set to a value. Below it, the text display shows "Celsius: {\$1:fixed=0}" and "Fahrenheit: {\$2:fixed=0}". To the right, a binding panel shows "Binding < {\$2} > + -", "Realm", "TempConverter", "variable", "Fahrenheit", and "value".

Здесь число дробных цифр установлено равным нулю, и мы видим округлённые до целых значения температур

The image shows the control interface with the slider and text display. The text display now shows "Celsius: -6" and "Fahrenheit: 21".

Значения времени, такие как временная позиция для видео или временная шкала WATCHOUT, могут быть отформатированы с использованием форматтера *show*, за которым следует нужная комбинация *hmsf*, представляющая часы, минуты, секунды, доли секунд. Например, следующая спецификация форматирования покажет минуты и секунды:

`{$1:show=ms}`

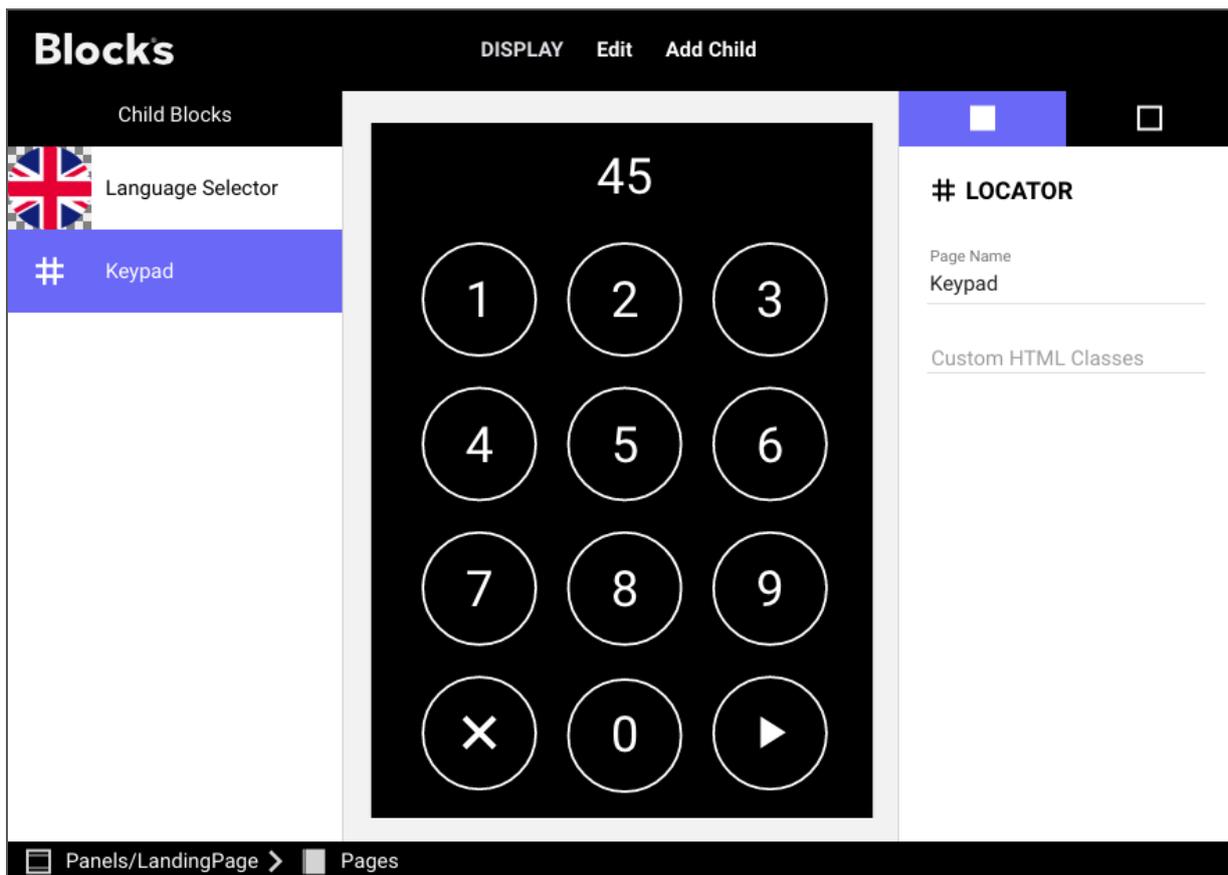
Locator (локатор)

Блок локатор (Locator) отличается от других типов блоков. Он не отображает собственного контента и не имеет дочерних блоков. При его появлении всё, что вы увидите, это цифровая клавиатура. Назначение этой клавиатуры – дать вам возможность указать, *где вы находитесь*, чтобы представить контент, связанный с конкретным спотом. Для вызова контента посетителю на смартфоне нужно набрать номер экспоната. При этом отображается блок, назначенный споту с этим номером ([Location ID](#)).

В зависимости от типа спота, связанного с этим идентификатором (ID), контент может быть любым: текстом или изображением, интерактивной многостраничной презентацией, видео или звуком. В большинстве случаев используют только звуковой клип, возможно, сопровождаемый изображением, подтверждающим, что вы находитесь в нужном месте. Это можно использовать для создания аудиогuida, для воспроизведения звука, локализованного на том месте, где вы находитесь, или описания конкретного объекта.

Auto-close when Playback Stops (автоматическое закрытие при остановке воспроизведения)

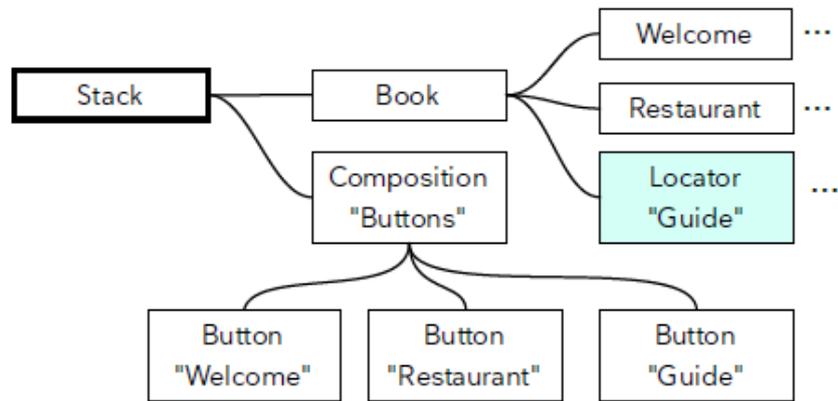
Этот параметр определяет, вернется ли локатор к цифровой клавиатуре, когда закончится воспроизведение (например, звука или видео). Как правило, это то, что вам нужно. Но если имеется несколько воспроизводимых по отдельности элементов контента, вы можете предпочесть разрешить воспроизведение и этого контента. В этом случае вам придется вручную вернуться к клавиатуре с помощью кнопки «Назад», показанной на целевой странице.



Блок Локатор (Locator) с цифровой клавиатурой.

Блок Locator можно использовать внутри других блоков для интерактивности. На рисунке (см. ниже) блок локатор находится внутри книги (Book), которая, в свою очередь, находится в блоке штабель (Stack). Кнопки, показанные внизу блока Stack, обеспечивают общую навигацию (как в примере, показанном в

разделе [Stack](#) этой главы). Страницы книги предоставляют разнообразный контент, в том числе композицию с кнопками для выбора языка и веб-блок со спецпредложениями в местном ресторане.



Локатор как часть интерактивного многостраничного мобильного гида.

На странице гида (Guide page) в приведенном выше примере отображается цифровая клавиатура. После ввода номера, спот с нужным ID будет обнаружен, и его блок появится на мобильном устройстве. Если это спот [Location](#) (местоположение), вы получите информацию (включая звук и видео), связанную с объектом в этом месте. Если это дисплей-спот ([Display spot](#)) или WATCHOUT-спот, вы услышите синхронизированный с видео звук с повествованием на выбранном вами языке.

Когда контент, назначенный споту, заканчивается, цифровая клавиатура появляется снова. Кнопка накладывается на контент и во время воспроизведения, что позволяет остановить воспроизведение в любой момент, вернувшись к цифровой клавиатуре.

Дизайн блоков для дисплей-спотов (Display) и гостевых спотов (Visitor)

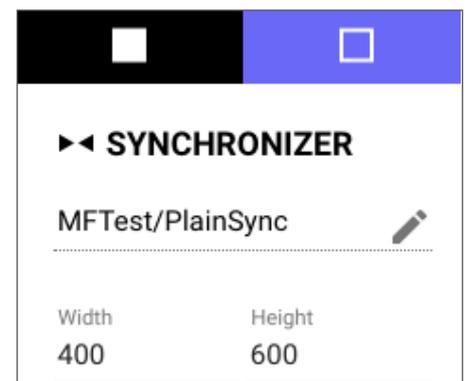
Контент для дисплей-спотов и для гостевых спотов (через блок Locator), должен быть создан с учётом размеров экранов. Блок выбора тэгов Tag Selector (или блок Synchronizer - синхронизатор) часто используется как самый верхний блок. Первый дочерний блок настроен, чтобы появиться на споте Display. Последующие дочерние блоки помечены как “mobile”, и они отображаются на гостевых спотах (по умолчанию для всех гостевых спотов добавлен тэг “mobile”). Подробнее – далее в разделе Синхронизатор Synchronizer.

Synchronizer (синхронизатор)

Блок Синхронизатор (Synchronizer) позволяет синхронизировать контент, например видео и звук, между спотами. Наиболее распространенный вариант – синхронизация гостевого спота (Visitor spot) с дисплей-спотом (Display spot) или WATCHOUT-спотом, а также синхронизация видео на нескольких дисплей-спотах. Помимо синхронизации, этот блок имеет те же возможности сопоставления тэгов, что и блок выбора тэгов (Tag Selector), который используется для выбора правильной звуковой дорожки согласно предпочтениям пользователей.

Чтобы воспроизведение было синхронным с дисплей-спотом или WATCHOUT-спотом, нужно сделать следующее:

1. Создайте блок Synchronizer, придав ему размеры, соответствующие мобильному устройству, как показано в примере справа.
2. Перетащите блок Synchronizer на дисплей-спот или WATCHOUT-спот и присвойте этому месту идентификатор местоположения ([Location ID](#)).
3. Поместите видеоролик на дисплей-спот в качестве первого дочернего блока синхронизатора, как показано на рисунке ниже. Переопределите



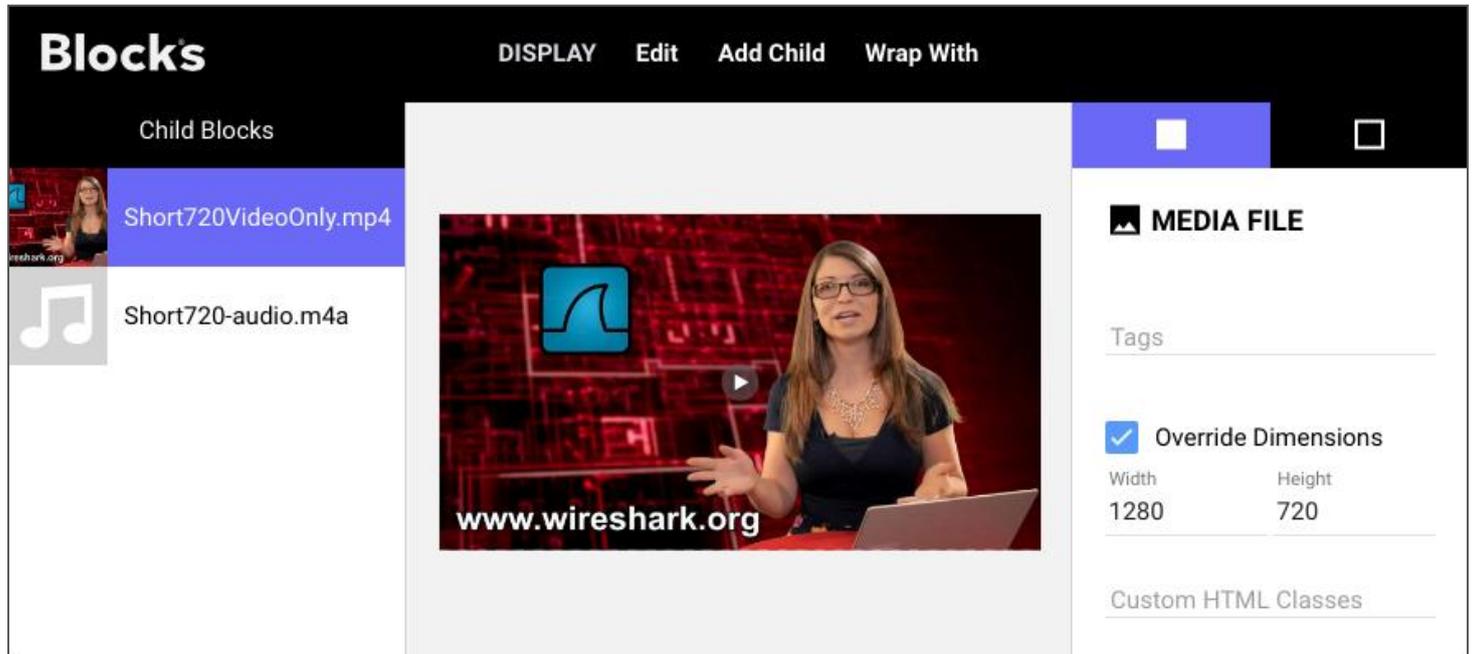
размеры этого первого дочернего блока, чтобы они соответствовали Display-споту (этот шаг неприменим для WATCHOUT-спотов).

4. Добавьте один или несколько звуковых медиа-блоков в качестве дочерних блоков синхронизатора (Synchronizer), назначив соответствующие тэги (см. ниже, на втором рисунке).

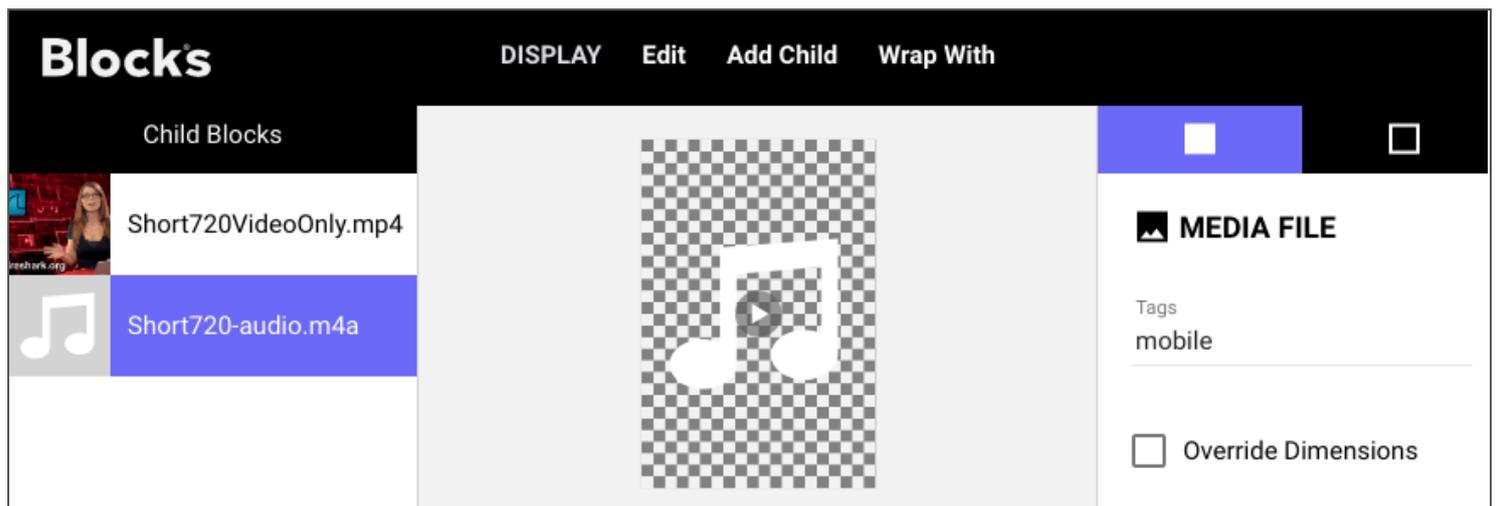
5. Свяжите блок Locator с гостевым спотом (Visitor).

6. С помощью цифровой клавиатуры на мобильном устройстве введите ID дисплей-спота.

Звук будет воспроизводиться через мобильное устройство синхронно с видео. Чтобы вместе со звуком на мобильном устройстве появлялся и визуальный контент, сначала добавьте композицию (Composition), а затем добавьте в нее звук и этот контент.



Добавьте видео первым дочерним блоком синхронизатора при использовании с дисплей-спотом. Измените размеры этого блока для соответствия целевому дисплей-споту.



Добавьте один или несколько звуковых блоков в синхронизатор, назначив тэг mobile, а также языковые тэги, если это необходимо.

Синхронизация видео на нескольких дисплеях

Синхронизатор (Synchronizer) можно использовать и для синхронизации воспроизведения видео на нескольких дисплеях. Эти ролики должны иметь одинаковую продолжительность и, как вариант, могут

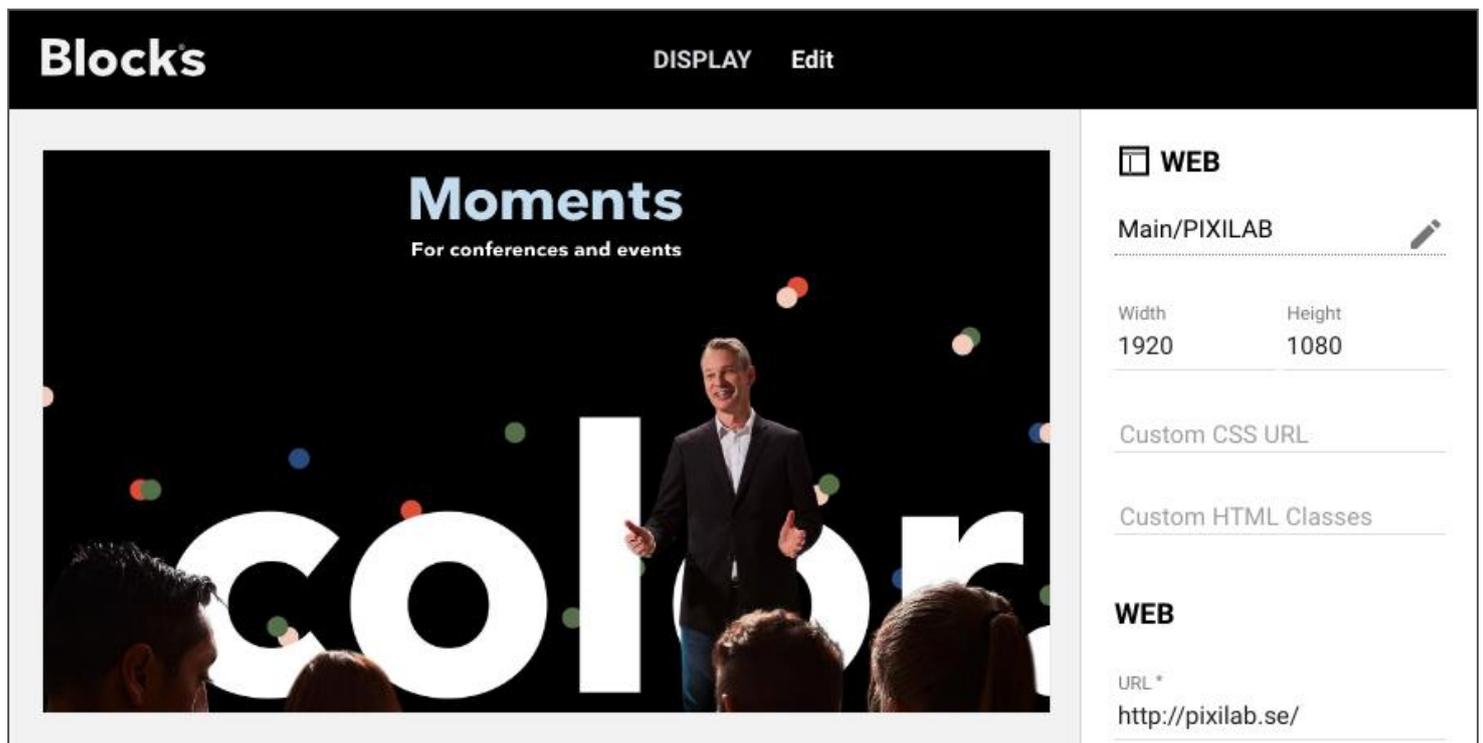
быть составными частями видео высокого разрешения, отделёнными от целого в соответствии с расположением дисплеев.

Используйте теги, чтобы связать видео с его монитором, назначив один и тот же тег дисплей-споту (Display) и соответствующему дочернему блоку синхронизатора (Synchronizer). Для трех мониторов, установленных рядом, вы можете использовать теги left, center, right (слева, по центру, справа). Обратите внимание, что вам не нужно добавлять блок выбора тегов (Tag Selector), чтобы это работало, так как синхронизатор (Synchronizer) имеет такую же способность выбора тегов.

▲ Такая синхронизация использует сервис *best-effort delivery*. Достигнутая точность синхронизации будет зависеть от типа используемых устройств воспроизведения и от продолжительности видео. Если требуется абсолютная точность, до кадра, вы можете использовать специализированное решение, такое как [Dataton WATCHOUT](#), которым затем можно будет управлять с помощью Blocks.

Блок Web

Добавьте блок Web, чтобы показать веб-сайт. Это может быть блок верхнего уровня или дочерний блок внутри композиции, слайд-шоу или внутри любого другого блока-вместителя.



The screenshot shows the Blocks interface. At the top, there's a header with 'Blocks' on the left and 'DISPLAY Edit' on the right. The main content area is a video player showing a man on stage with the word 'color' in large white letters. To the right of the video player is a configuration panel for the 'WEB' block. The panel has a 'WEB' header with a small icon and a pencil icon. Below that, it shows 'Main/PIXILAB' with a pencil icon. There are two input fields for 'Width' (1920) and 'Height' (1080). Below these are two more input fields: 'Custom CSS URL' and 'Custom HTML Classes'. At the bottom of the panel, there's a 'WEB' section with a 'URL*' field containing 'http://pixilab.se/'.

Блок Web как блок верхнего уровня.

▲ Имейте в виду, что для загрузки веб-сайтов может потребоваться некоторое время. Если веб-блок используется в слайд-шоу, он начнет загружаться при отображении предыдущего блока слайд-шоу. Следовательно, целесообразно, чтобы блок, предшествующий веб-блоку, оставался на экране достаточно долго, чтобы веб-блок успел загрузиться.

Ограничения

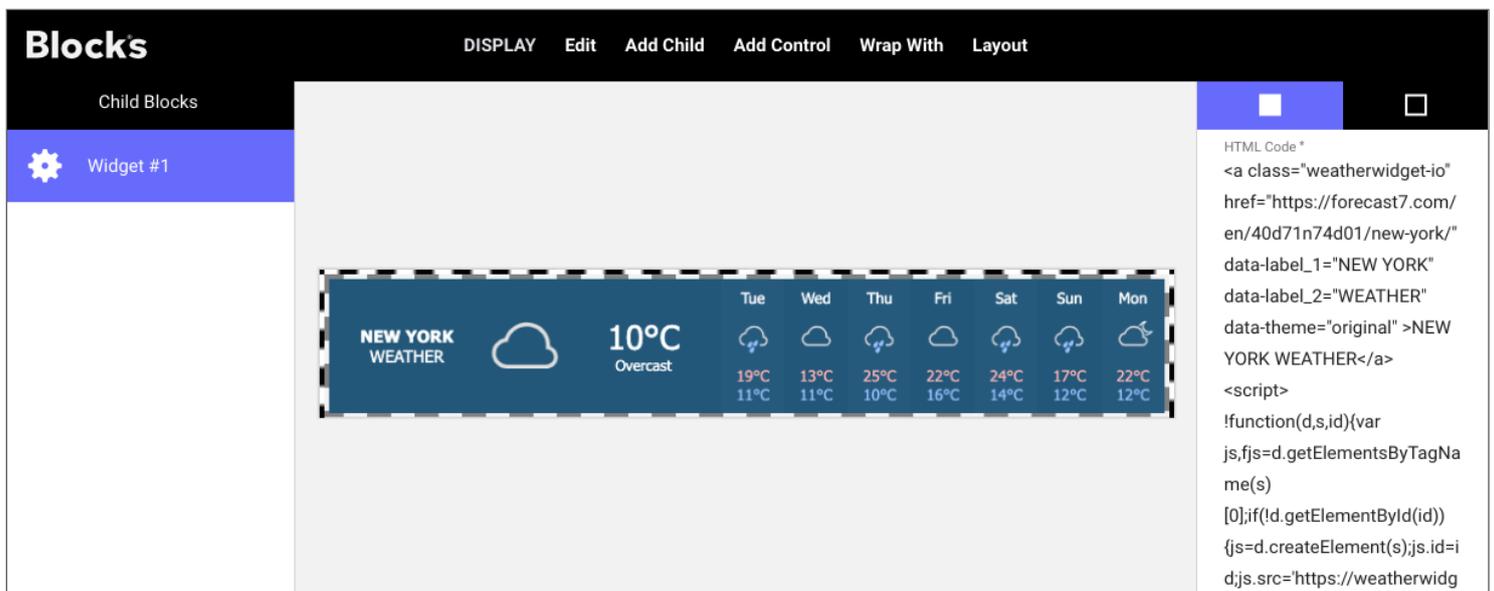
Некоторые сайты не совместимы с веб-блоком. Эти сайты предупреждают, что они не могут быть встроены как часть других веб-решений, установив «X-Frame-Option». В некоторых случаях вы можете обойти это с помощью расширений браузера, таких как [xframe-assassin](#) или [iframe-allow](#) для Chrome.

Кроме того, если по какой-то причине вы используете свой спот с применением безопасного соединения (https), веб-сайты, использующие простой протокол HTTP, не могут быть показаны - только сайты, которые также используют HTTPS.

▲ Для работы Blocks по безопасному соединению требуется сертификат и специальная конфигурация сервера, что выходит за рамки этого документа. Свяжитесь с PIXILAB для получения более подробной информации, если вам нужно это сделать.

Widget

Этот тип блока аналогичен веб-блоку, описанному выше. Однако вместо URL-адреса, указывающего на веб-контент, блок Widget размещает контент внутри себя. Этот метод обычно используется для отображения динамического контента, такого как карты погоды или часы.



The screenshot shows the Blocks editor interface. At the top, there are tabs for 'DISPLAY', 'Edit', 'Add Child', 'Add Control', 'Wrap With', and 'Layout'. Below the tabs, there's a 'Child Blocks' section with a 'Widget #1' block selected. The main area shows a weather widget for New York with a temperature of 10°C and a forecast for the next seven days. The right sidebar shows the HTML code for the widget:

```
HTML Code *
<a class="weatherwidget-io"
href="https://forecast7.com/
en/40d71n74d01/new-york/"
data-label_1="NEW YORK"
data-label_2="WEATHER"
data-theme="original" >NEW
YORK WEATHER</a>
<script>
!function(d,s,id){var
js,fjs=d.getElementsByTagName(s)
[0];if(!d.getElementById(id))
{js=d.createElement(s);js.id=i
djs.src="https://weatherwidg
```

Этот виджет погоды мы нашли на сайте <https://weatherwidget.io>

Код виджета, полученный с этого сайта, был вставлен в поле “HTML Code”, см. в правой части рисунка. На нашем Blocks wiki можно узнать больше об использовании виджетов: https://int.pixilab.se/docs/blocks/signage#using_widgets

Reference (ссылка)

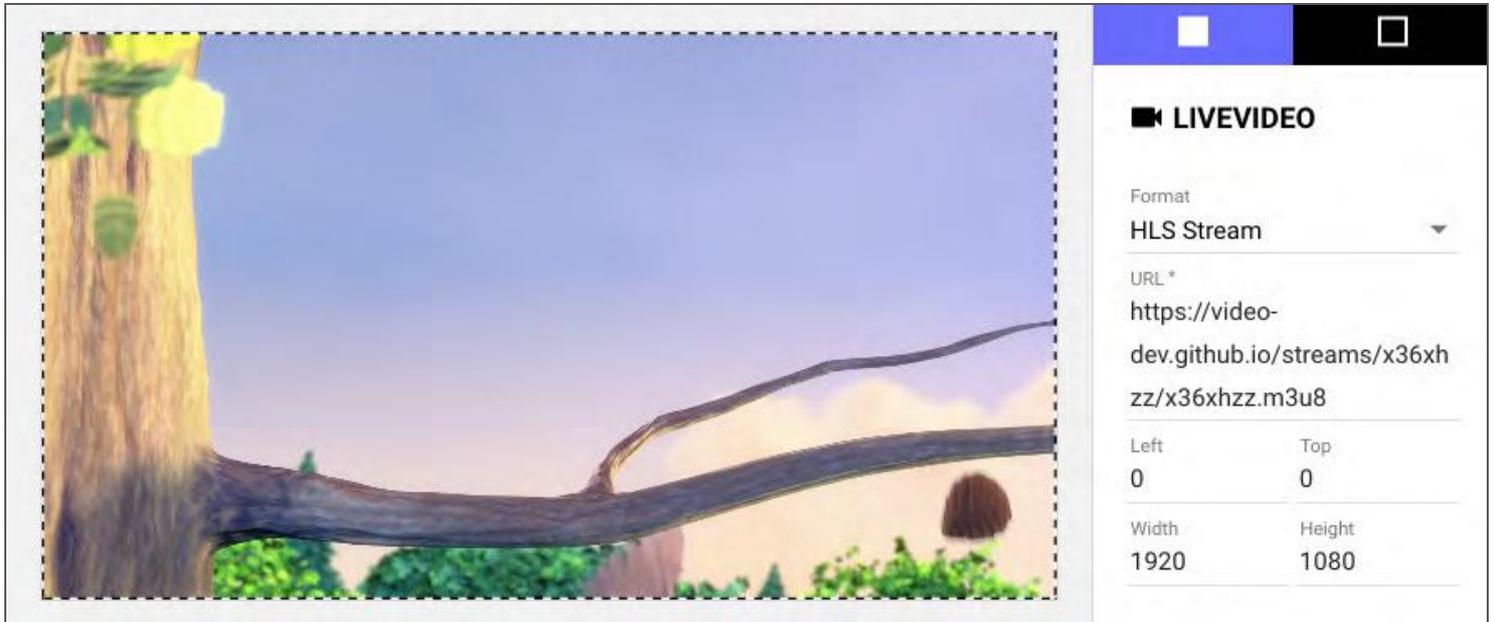
Используйте блок Reference, чтобы ввести весь блок верхнего уровня как дочерний элемент другого блока. Это особенно полезно для контента, который отображается в нескольких местах. Вместо того, чтобы копировать один и тот же контент или блок во все эти места, сделайте следующее:

1. Добавьте общий контент на страницу Display как отдельный блок верхнего уровня (например, как медиафайл или композицию – [Media File](#) или [Composition](#)).
2. Включите блок верхнего уровня, где это необходимо, используя блок Reference, указав нужный блок как “Referenced group/block”.

Совместное использование единственного экземпляра часто используемого контента облегчает обновление этого контента, если возникнет такая необходимость, избавляя от необходимости замены этого контента в нескольких местах.

Live Video

Используйте блок Live Video для отображения потокового видео, исходящего с камеры, веб-службы или другого подходящего источника. Как и веб-блок, описанный выше, все, что вам нужно, это URL-адрес видеопотока.



Форматы живого видео

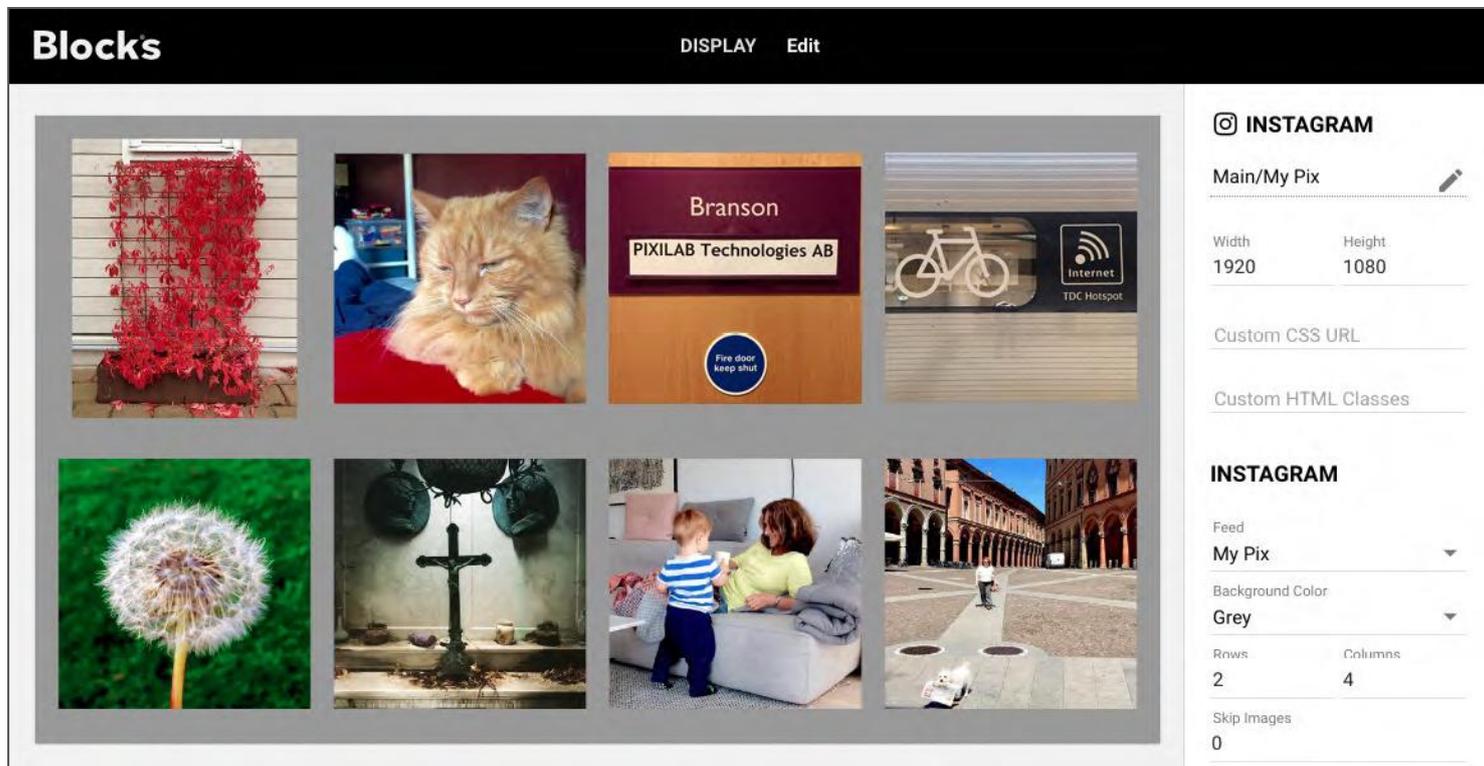
Выберите ожидаемый тип видеопотока в разделе Format:

- **HLS stream:** совместим с большинством плееров. Поддерживает звук. Задержка - несколько секунд. Поддерживается некоторыми потоковыми боксами и большинством веб-сервисов.
- **MJPEG stream.** Совместим с большинством плееров. Нет звука. Малая задержка. Поддерживается многими видеокамерами.
- **JPEG polling.** Совместим со всеми плеерами. Нет звука. Малая задержка. Поддерживается некоторыми видеокамерами.
- **RTSP stream.** Совместим только с плеерами Brightsign. Поддерживает звук. Умеренная задержка. Поддерживается большинством видеокамер и потоковых боксов.

На нашем Blocks wiki можно узнать больше о том, как использовать живое видео <https://int.pixilab.se/docs/blocks/live-video> .

Instagram

Используйте блок Instagram, чтобы показать изображения из ленты Instagram.. Это может быть блок верхнего уровня или дочерний блок внутри композиции, слайд-шоу или любого другого блока-вместителя. Новые изображения появляются на экране автоматически по мере добавления их на канал.



Blocks DISPLAY Edit

Instagram

Main/My Pix 

Width	Height
1920	1080

Custom CSS URL

Custom HTML Classes

INSTAGRAM

Feed
My Pix 

Background Color
Grey 

Rrws	Columns
2	4

Skip Images
0

Отображение картинок из Instagram.

Feed (загрузка)

Имя загрузки. Загрузки настроены в разделе [Instagram Feeds](#) на странице Manage (управление).

Background Color (цвет фона)

Цвет, который будет подложкой для изображений. Можно выбрать опцию «белый» (white), так как некоторые картинки могут быть на белом фоне.

Rows/Columns (строки / столбцы)

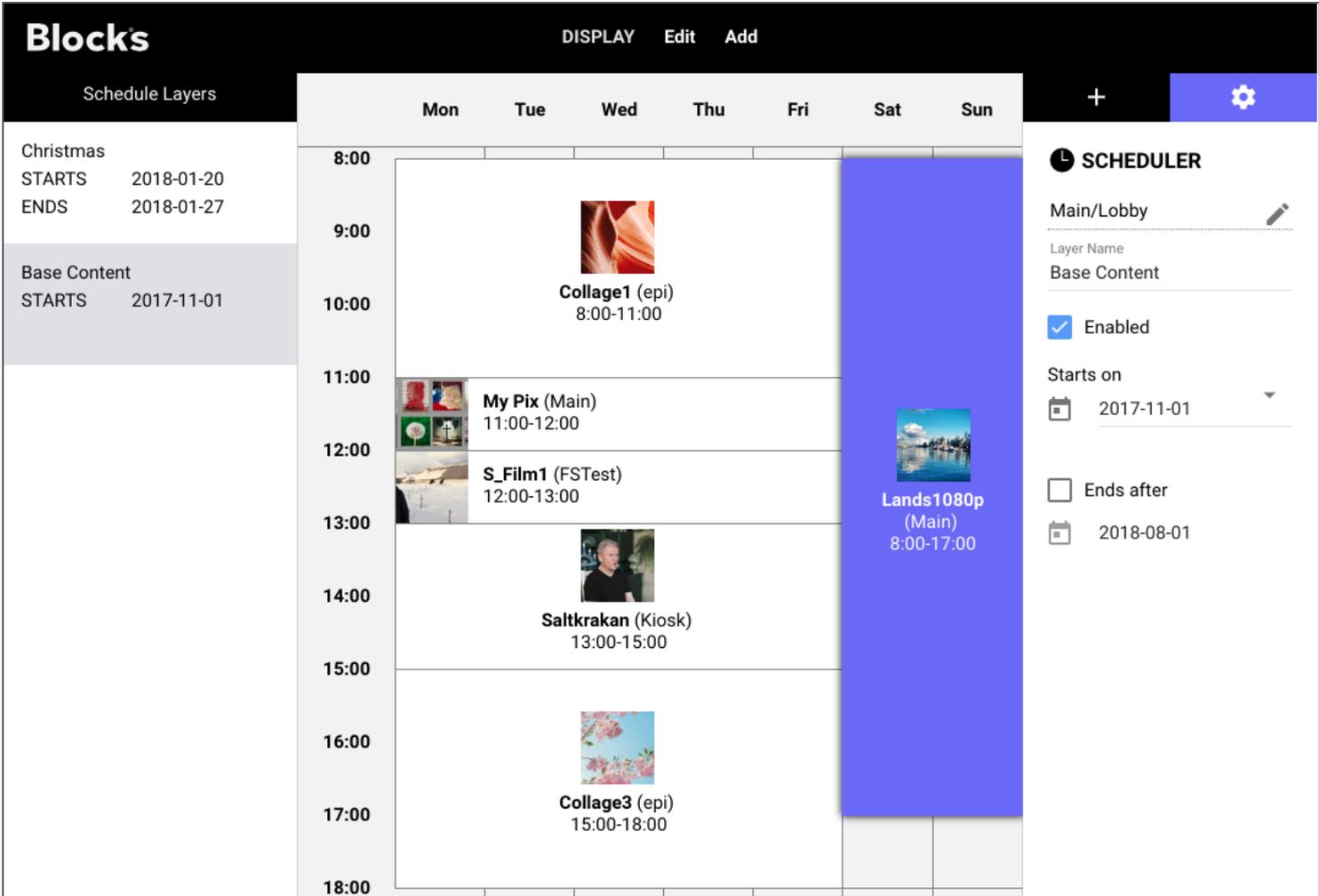
Сколько картинок нужно показывать. В приведенном выше примере показаны две строки с четырьмя изображениями в каждой, в общей сложности – восемь изображений.

Skip Images (пропустить изображения)

Пропустите указанное количество изображений в начале загрузки. Это полезно при показе изображений из одной и той же загрузки на нескольких соседних дисплей-спотах (Display). Установите смещение для блока, отображаемого на первом дисплее, на 0, для второго дисплея - на 8 (предполагается, что показано 8 изображений на каждом дисплее) и т.д.

Schedule (расписание)

Этот тип блоков позволяет показывать другие блоки в заранее определённые моменты времени или заданные даты. Расписание назначается споту, как и любой другой блок. Однако расписание не может использоваться внутри других блоков. Оно может быть только блоком верхнего уровня.



Разный контент для будней и выходных дней.

Чтобы добавить контент в расписание, нажмите кнопку «плюс» в верхнем правом углу. Найдите нужный блок и перетащите его в расписание. Если в вашей системе много блоков, их можно найти с помощью фильтра по имени их группы, что облегчает поиск.

После того, как блок добавлен в расписание, его можно перемещать и изменять, чтобы определить, когда он должен появиться.

Слои расписания (Schedule Layers)

Контент, добавленный в расписание, располагается слоями. Первоначально существует только один слой. Чтобы добавить слой, выберите “Layer” в меню Add (добавить). Слой можно отключить, сняв флажок “Enabled” (включено). Отключенные слои не воспроизводятся. Таким же способом можно ограничить воспроизведение слоя датами, используя опции “Starts on” (начало) и “Ends after” (окончание после). Снимите флажок “Ends after”, чтобы не указывать дату окончания.

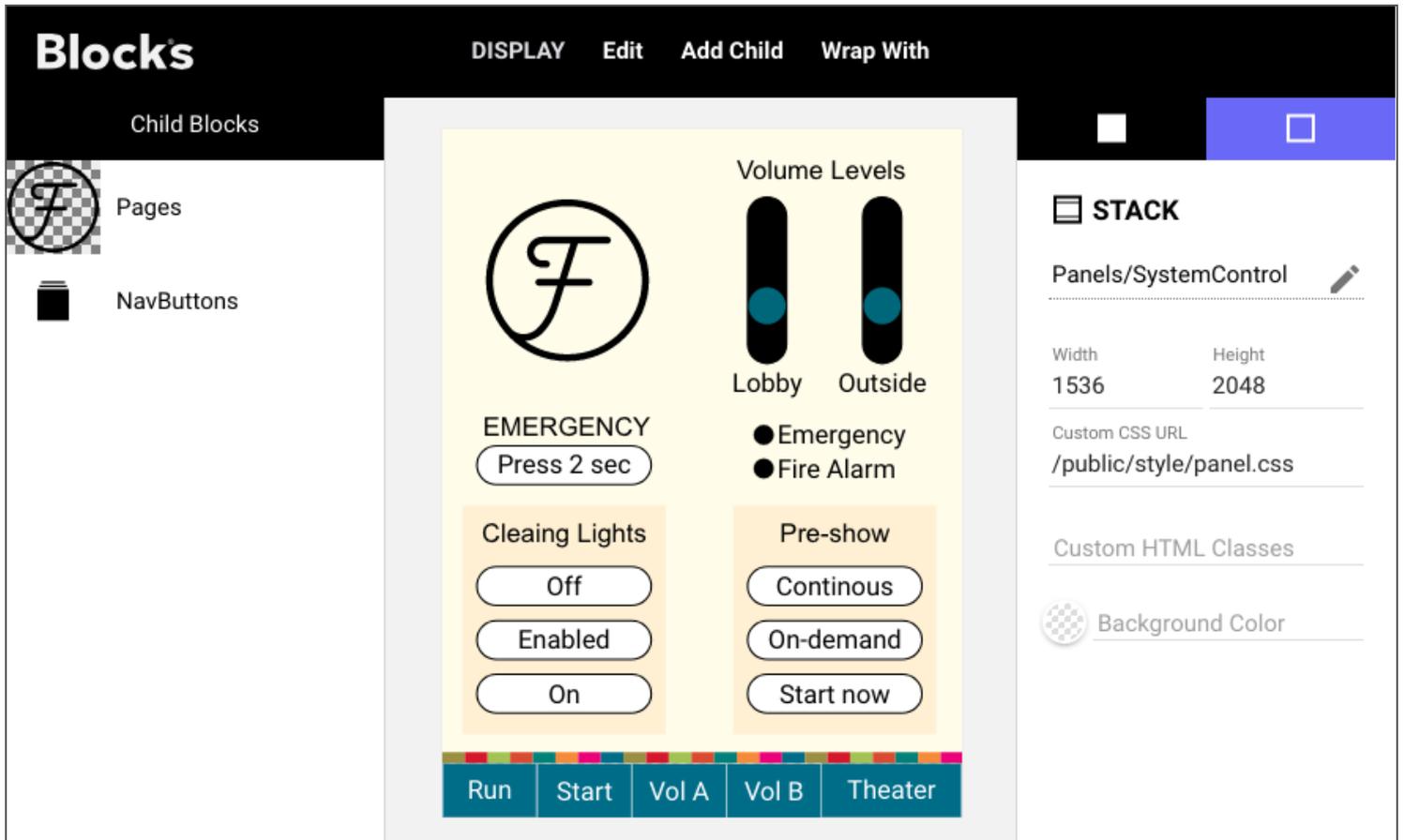
Используйте дополнительные слои для планирования альтернативного контента, сохраняя при этом основной контент. Система будет воспроизводить контент с верхнего включенного слоя, который находится в пределах своего диапазона дат и имеет контент для текущего времени и дня недели. Если в верхнем слое нет контента, отвечающего этим критериям, будет рассмотрен нижележащий слой и т.д.

The screenshot displays the 'Blocks' Scheduler interface. On the left, the 'Schedule Layers' list shows two layers: 'Christmas' (STARTS: 2018-01-20, ENDS: 2018-01-27) and 'Base Content' (STARTS: 2017-11-01). The main grid shows a weekly schedule from Monday to Sunday, with time slots from 10:00 to 15:00. A blue block for 'Christmas' is scheduled from 11:00 to 14:00 on Thursday. The block contains a thumbnail image of a woman and the text 'P1 (Panels) 11:00-14:00'. On the right, the 'SCHEDULER' settings panel for 'Main/Lobby' shows the layer name 'Christmas', 'Enabled' checked, 'Starts on' 2017-12-20, and 'Ends after' 2017-12-27.

Блок слоя Christmas будет воспроизводиться с 11:00 до 14:00 между указанными датами вместо блоков основного слоя. Контент основного слоя будет продолжат воспроизводиться в другое время.

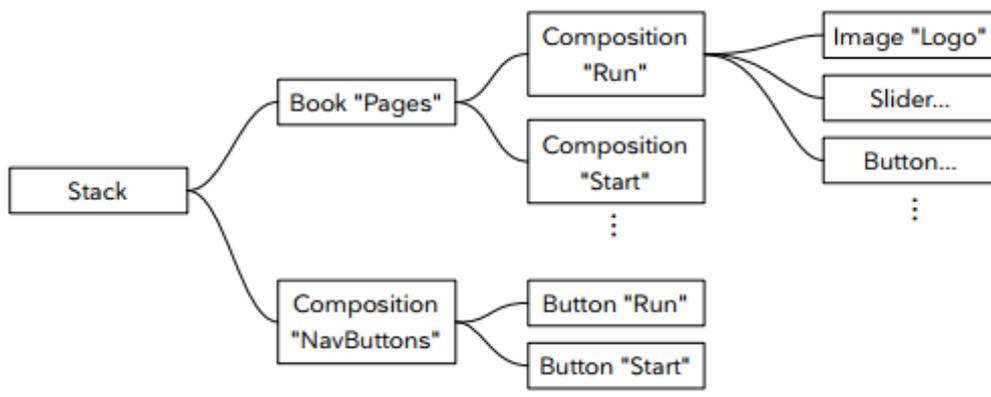
6. ПАНЕЛИ УПРАВЛЕНИЯ

Для создания панелей с кнопками и другими элементами управления используют блок композиции ([Composition](#)). Композиция может включать и другие типы блоков. Это может быть использовано для создания интерактивных киосков для посетителей, а также панелей управления персоналом (например, для управления воспроизведением на дисплеях, для включения питания, для регулирования освещения, уровнях громкости и т. д.)



Панель управления с кнопками и движками. Синие кнопки на панели навигации внизу имеют пользовательский стиль.

Кнопки и другие элементы управления были добавлены в композицию при помощи меню [“Add Control”](#). Ниже – блоковая структура для этой панели.



Панель управления на основе блока *Stack* с кнопками навигации по нижнему краю (“Stick to Edge”).

Button (кнопка)



Используйте кнопку для управления некоторыми состояниями системы, для навигации в интерактивном дизайне или для интерактивной привязки тэгов к споту.

The screenshot shows the 'Blocks' software interface. On the left is a 'Child Blocks' panel with various controls like 'On', 'Pre-show', 'Cleaning Lights', 'Enabled', 'Fire Alarm', and 'Start now' (which is selected). The main workspace displays a control panel with a large 'F' logo, 'EMERGENCY Press 2 sec' button, 'Cleaning Lights' section with 'Enabled' and 'On' buttons, and 'Volume Levels' section with 'Lobby' and 'Outside' sliders and 'Emergency'/'Fire Alarm' buttons. The 'Pre-show' section has 'Continous', 'On-demand', and 'Start now' buttons. On the right, a 'BUTTON' configuration panel is open, showing settings for 'Title' (Start now), 'Title Position' (Inside), dimensions (Left: 900, Top: 1620, Width: 520, Height: 120), 'Behavior' (System selected), 'Target' (WATCHOUT, WO1), and 'playing' status (true).

Кнопка настроена на запуск шоу WATCHOUT.

Title /Title Position (надпись / положение надписи)

Задаёт название кнопки, а также положение, в котором это название должно появляться на кнопке или вне её (или вообще будет отсутствовать).

Position (расположение)

Позволяет вам численно задавать положение и размеры кнопки. Вы также можете перетаскивать кнопку или один из ее краев мышью. Включите функцию «[Snap to Grid](#)» (привязать к сетке) в настройках внешнего блока, чтобы упростить выстраивание элементов. Вы также можете использовать команды из меню [Layout](#) (макет) для размещения нескольких кнопок или других элементов управления по горизонтали или по вертикали.

Behaviors (поведение кнопки)

Указывает, будет ли кнопка управлять системной функцией (например, “play a video on a spot” или “turn on the lights”) или локальной функцией, влияющей только на конкретный спот (например, перейти на другую страницу или установить тэг). Этот параметр определяет то, что появится следом. Изначально кнопка имеет единственное поведение. Чтобы добавить «поведение», нажмите символ +. Используйте стрелки для перемещения между поведением, ориентируйтесь на цифры между стрелками. Внешний вид кнопки определяется ее первым поведением.

System Behaviors (режимы системы)

После выбора “System” в качестве режима кнопки (Button Behavior) вы подсоединяете кнопку к целевому свойству, используя поля, показанные ниже. В примере, показанном выше, выбранная кнопка подсоединяется к состоянию «воспроизведение» (playing) кластера WATCHOUT с именем «PreShow».

Параметры, задаваемые в разделе Target (цель), зависят от выбранного объекта и типа выбранного свойства. В приведенном выше примере свойство «воспроизведение» (playing) является истинным/ложным (логическим, булевым), значение отображается внизу. Последний пункт содержит следующие опции:

- **Set to (задано...)**. Задаёт значение свойства при нажатии кнопки.
- **Set while pressed (задано, пока нажимаешь)**. Устанавливает одно значение при нажатии кнопки, а другое - при её отпускании. Это приводит к мгновенному действию, которое возвращается к альтернативному значению, как только вы отпускаете кнопку.
- **Toggle (переключение)**. Изменяет состояние между двумя возможными значениями каждый раз, когда нажимается кнопка. Это можно использовать, чтобы одна и та же кнопка что-то включала (ON) или выключала (OFF).
- **Increase by (увеличить на...)** и **Decrease by (уменьшить на...)**. Добавляет (или вычитает) число к свойству. Можно использовать для увеличения/уменьшения уровня освещённости или аналогичного параметра при каждом нажатии кнопки (на фиксированный процент). Эта опция доступна только для свойств, имеющих численное значение.

Типы свойств

Значения, которые могут быть заданы кнопкой, зависят от типа выбранного свойства. Доступны следующие типы свойств:

- **Boolean (логические (булевы))**. Доступны только два состояния: *истина и ложь (true и false)*. Такой вариант для свойства «воспроизведение» показан на рисунке выше.
- **Number (числовые)**. Числовое значение, которое в некоторых случаях может быть отрицательным и может содержать дробную часть. В примерах представлены уровень громкости дисплей-спота (Display spot) и значение “Input” (вход) для подсистемы WATCHOUT.
- **Time (временные)**. Положение во времени, выраженное в формате (НН: ММ.СС.fff), причем единицы измерения, принятые по умолчанию – секунды. Пример – положение во времени на временной шкале WATCHOUT.
- **String (строка)**. Текстовое значение, например имя блока, назначенного споту.

См. главу [«Свойства и функции»](#), в которой подробно описаны свойства, доступные для кнопок.

Local Behaviors (локальные поведения)

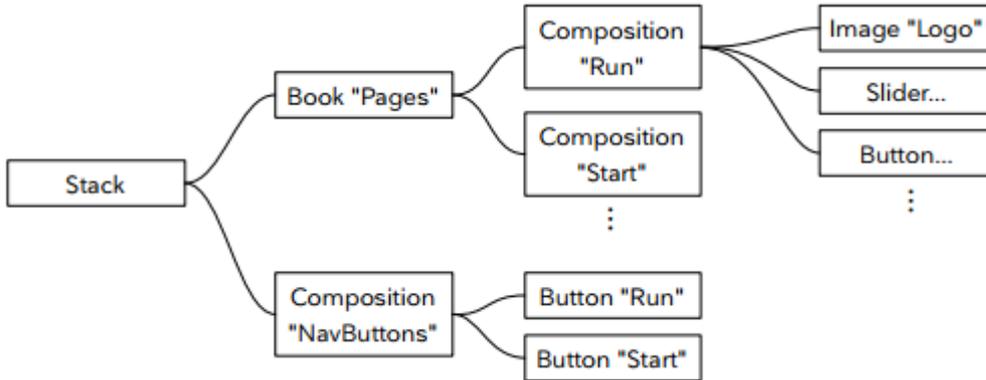
Локальные поведения влияют только на дисплей-спот (Display) или на гостевой спот (Visitor), к которым они применяются.

- **Go to Page (перейти на страницу)**. Переход на другую страницу в книге (Book). Подробности смотрите ниже.
- **Toggle/Select/Reset Tags(переключить/выбрать/сбросить тэги)**. Тэги, специфичные для спотов, добавляются или удаляются, как описано ниже в разделе [«Установка / удаление тэгов»](#).

- **Set a Spot Parameter (установить параметр спота).** Параметры могут использоваться в путях свойств, чтобы можно было назначить один и тот же блок для разных устройств (см. [«Параметризация путей свойств» \(Parametrization of Property Paths\)](#)).

Поиск целевой страницы по имени (Locating a Target Page by Name)

Когда выбран режим «Локальное поведение» (Local Behavior) и задано «Перейти на страницу» (Go to Page), вы указываете целевую страницу в текстовой строке. Этим вы указываете путь к целевой странице, начиная либо со страницы, содержащей кнопку (относительный путь), либо от самого верхнего блока (абсолютный путь).



Например, если вы хотите, чтобы кнопка «Старт» (Start), показанная на рисунке выше, открывала страницу «Пуск» (Start) в Книге (Book), вы можете указать путь одним из следующих способов:

- **“/Pages/Start”.** Указывается абсолютный путь, начиная с блока Стек (Stack) верхнего уровня, который содержит Книгу под названием «Страницы» (Pages), которая, в свою очередь, имеет композицию с именем «Старт» (Start).
- **“../Pages/Start”.** Это относительный путь, начиная со страницы, содержащей кнопку. Кнопка находится в композиции «NavButtons» (кнопки навигации) в блоке Стек (Stack) верхнего уровня. Две точки (..) означают «подняться на один уровень» (“go up one level”), что приводит вас к блоку Стек (Stack) верхнего уровня, из которого вы затем переходите в «Страницы» (Pages) и «Старт» (Start).

В этом случае абсолютная форма является предпочтительной, так как она и короче, и проще для понимания. Однако в более сложных и более глубоких блочных иерархиях часто бывает проще использовать относительные пути.

▲ Абсолютный путь начинается с косой черты, а относительный путь – нет. Синтаксис пути напоминает пути файловой системы в операционных системах на базе Unix, таких как macOS.

Поиск цели с помощью чисел или пошагово (Locating a Target Numerically or Incrementally)

Для поиска можно использовать не только имя, но и числовые значения цели: просто число вместо имени, а первый дочерний блок имеет номер 0 (ноль). В приведенном выше примере вы можете перейти на вторую страницу книги «Pages», указав путь «/ Pages / 1» (где 1 – вторая страница, поскольку первая страница имеет номер 0).

▲ Не обозначайте страницу только цифрами, если вы хотите перейти на эту страницу по имени. Если вы хотите использовать здесь число, добавьте к нему какой-нибудь текст, например «page2», «page3» и т. д.

Аналогичный синтаксис можно использовать для постепенного перемещения вперед или назад на указанное количество страниц. Чтобы в книге перейти на две страницы вперед, укажите «/ Pages / + 2» или «/ Pages / -1», чтобы вернуться на одну страницу назад. Этот синтаксис может быть использован для

перехода вперед или назад, но не сможет последней страницы / слайда.

преодолеть границу, достигнув первой или

При использовании только + или - (без последующих цифр) происходит перемещение на один шаг в слайд-шоу или в книге, в том числе от последней страницы к первой и наоборот. При перемещении вперед по слайд-шоу можно применять любые переходы (transitions).

Поиск целей внутри блока Reference (ссылка)

При использовании блока [Reference \(ссылка\)](#) для перехода в блок верхнего уровня вы можете перемещаться внутри ссылочного блока любым из методов, описанных выше. Однако если вам нужно нацелиться на блок вне ссылочного блока, придётся использовать немного другой синтаксис. Это связано с тем, что абсолютный целевой синтаксис (начиная с одиночной косой черты, как обсуждалось выше) работает на основе корня ссылочного блока – так же, как если бы ссылочный блок использовался сам по себе.

Чтобы вырваться из ссылочного блока, вплоть до конечного корня самого внешнего блока в ссылочной цепочке, используйте абсолютный синтаксис, который начинается с двух косых черт //. Затем вы можете перемещаться вниз оттуда к обычным дочерним блокам или ссылочным блокам (которые могут в свою очередь ссылаться на другие блоки на нескольких уровнях).

▲ Синтаксис с двойной косой чертой не нужен при поиске целей с помощью оператора задачи, так как такие ссылки всегда начинаются с первичного корневого уровня.

Setting/Clearing Tags (установка / удаление тегов)

Эти опции используются для добавления / удаления тегов. Тег – это отдельное слово или несколько слов, которые влияют на поведение спота.

Можно назначить несколько тегов, разделяя их запятой. Доступны следующие опции:

- **Toggle Tags.** Переключить теги. Нажатие кнопки добавит тег (и), если он еще не установлен, или удалит установленный тег (и).
- **Reset Tags.** Сбросить теги. Нажатие на кнопку удалит указанный тег (и).
- **Select Tags.** Выберите теги. Устанавливает указанные теги из набора тегов. Т.е. будут установлены теги, выбранные в поле «Tags», а теги, указанные в поле «Of Tag Set», будут отключены. Например, чтобы выбрать языковой тег *swe* из набора *swe, eng, ger*, введите первый (*swe*) в поле «Tags» и укажите *swe, eng, ger* в поле «Of Tag Set». В поле «Of Tag Set» мы указываем все теги, которые могут применяться к файлу, а в поле «Tags» указываем тот тег, который назначаем в конкретном условии.

Теги могут быть назначены для спота с самого начала в поле “[Tags](#)” в диалоговом окне спота. Кроме того, теги могут быть применены к споту посредством кнопок, как описано выше, или через использование задач и скриптов. Набор тегов является локальным для спота и не влияет на другие споты.

Текущий набор тегов используют в сочетании с блоком [Tag Selector](#) или блоком [Synchronizer](#), чтобы выбрать дочерний блок для отображения. Это подходит для выбора языка и других аналогичных локальных настроек.

Setting a Spot Parameter (установка параметра спота)

Выберите «Set Spot Parameter», затем выберите или введите имя параметра и его значение. Подробнее об этом в главе “[Parametrization of Property Paths](#)” (параметризация путей свойств).

Button Feedback (обратная связь кнопки)

Кнопка загорается, чтобы указать состояние первого (или единственного) свойства или локальное поведение, которым она управляет. В частности, она загорается при следующих условиях:

- При управлении свойством Система (System) кнопка загорается, когда свойство имеет значение «Задано...» (Set to) или при первом из двух указанных значений, когда выбрано «Задано при нажатии» (Set while pressed) или «Переключение» (Toggle).
- Если установлено значение Local; Go to Page, кнопка загорится, когда указанная страница будет видна. Это особенно полезно для кнопок navigation bar, как в примере, показанном на рисунке выше, где вы можете одновременно видеть и кнопку, и целевую страницу.
- Если задано «Локальный» (Local); Выбрать/Переключить тэги (Select/Toggle Tags), кнопка загорится, когда установлены все указанные тэги.
- При установке на Local; Set Spot Parameter, он загорается, когда параметр имеет указанное значение.

▲ Такая индикация состояния не отображается в редакторе Blocks (Blocks editor), а только при взаимодействии с реальным дисплей-спотом/гостевым спотом (Display/Visitor spot).

Indicator (индикатор) ●

Индикатор похож на кнопку Button, см. выше, но обеспечивает только функции обратной связи и работает только в сочетании со свойствами Системы (System properties).

Slider (ползунок) —●—

Ползунок постепенно изменяет значение числового свойства. Он подходит для таких свойств, как громкость или яркость. Вы можете указать максимальные и минимальные значения, которые можно получить с помощью ползунка (что может быть меньше максимального диапазона, поддерживаемого этим свойством). Ползунок можно использовать с логическими свойствами (on/off), и тогда он работает как простой переключатель.

Подключите ползунок к нужному свойству так же, как это описано выше в разделе [«System Behaviors» \(режимы системы\)](#). Ползунок также показывает значение свойства, которым он управляет.

Bar (полоса) —■—

Полоса (Bar) подобна Ползунку (Slider), однако её невозможно использовать для управления свойством. Она предназначена только для отображения значения свойства.

Текст T

Добавляет как статический, так и динамический текст различных размеров и стилей. Вы можете использовать текстовый блок как самостоятельный блок верхнего уровня. Вы также можете использовать его внутри композиции. Если вы хотите использовать текст в других местах, сначала добавьте блок Composition, а затем добавьте блок Text внутри нее. Смотрите [“Text”](#) в главе под названием “Blocks” для более подробной информации.

Пользовательский стиль (Custom Styling)

Текст (а также кнопки и другие элементы управления) по умолчанию использует простой шрифт и стиль, но вы можете переопределить это с помощью каскадных таблиц стилей (CSS); как это сделать, смотрите на https://int.pixilab.se/docs/blocks/custom_styling.

Text Input (ввод текста)

T Используйте элемент управления вводом текста (Text Input) для ввода числовых значений или текста. Свяжите элемент управления Text Input с требуемым свойством, таким как переменная области (realm variable).

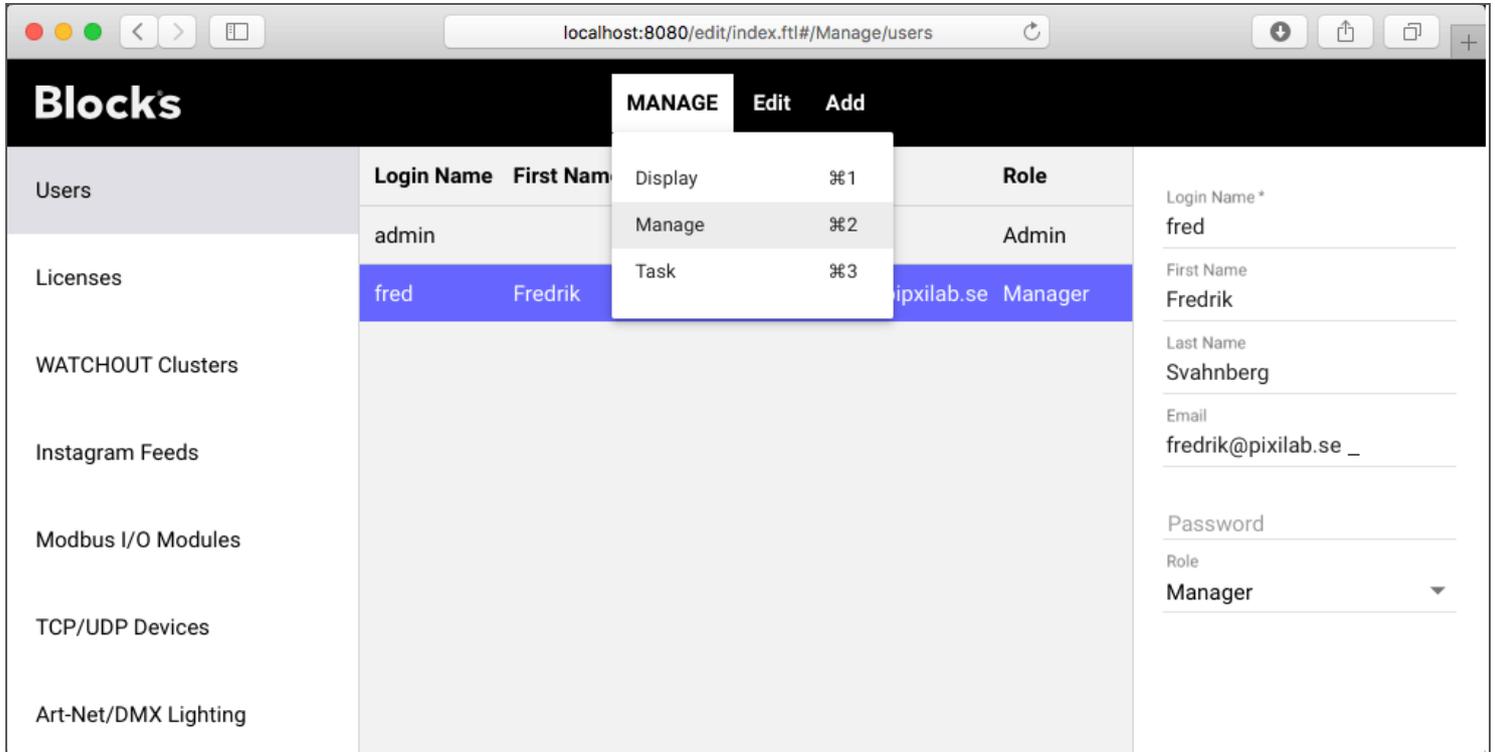
При привязке к числовому свойству получится вводить только цифры.

Установите флажок “Multiple Lines” (несколько строк), чтобы разрешить ввод нескольких строк текста.

Когда выбрано “Multiple Lines”, при нажатии Enter / Return при вводе значения создается новая строка, а не вводится значение. Чтобы просмотреть несколько строк текста, измените размер элемента управления по вертикали.

7. MANAGE (УПРАВЛЕНИЕ)

Страница Manage (управление) используется для включения в систему и для настройки объектов: пользователей, кластеров WATCHOUT, устройств ввода-вывода, сетевых устройств и других. Доступ к этой странице происходит командой Manage в верхнем меню. В списке слева выберите область для управления.



Выберите Manage в строке меню.

Users (пользователи)

В системе всегда присутствует как минимум один пользователь, который называется “admin”. Этот пользователь не может быть удален или переименован. Вы можете (и должны) изменить пароль для пользователя “admin”. Вы можете ввести его имя / фамилию и адрес электронной почты.

Другие пользователи могут быть добавлены через меню “Add” (добавить). Вы должны указать “Login Name” для входа в систему вместе с паролем. Вы можете указать дополнительную информацию для идентификации пользователя. Параметр Role указывает на функцию пользователя; пользователь “admin” имеет полный доступ к Blocks (включая добавление пользователей и установку их ролей).

См. [“Авторизация и роли”](#) для получения дополнительной информации о ролях и правах доступа.

Licenses (лицензии)

Наполнение вашего лицензионного ключа Blocks отображается в разделе “Licenses”. Слоты Display, Visitor, Location и WATCHOUT лицензируются по отдельности, что позволяет адаптировать лицензию к конкретным потребностям. Для каждой лицензионной функции вы можете увидеть

- Общее количество имеющихся лицензий.
- Количество слотов, которые задействованы в текущей конфигурации системы (слоты, которые мы добавили при конфигурировании шоу).

• Количество активных спотов из общего количества добавленных спотов (см. второй пункт). Это позволяет понять, какие устройства в данный момент не задействованы или по какой-то причине не загрузились.

▲ Для версии Blocks под Linux требуется дополнительная лицензия. Эта лицензия также дает право строить и разворачивать сервер на основе нашего предварительно сконфигурированного [образа Linux](#).

Кластеры WATCHOUT

Добавьте и настройте любое количество кластеров WATCHOUT. Программное обеспечение [WATCHOUT](#) иногда используется с Blocks для мультиэкранной проекции или системы дисплеев. Кластер WATCHOUT состоит из одного или нескольких дисплей-компьютеров и (необязательно) продакшн-компьютера. Продакшн-компьютер обычно используется только во время подготовки шоу и затем может быть удалён. Однако в некоторых случаях он остается частью системы, чтобы быстро вносить необходимые изменения.

Добавьте кластер дисплеев с помощью меню “Add” (добавить) после выбора “WATCHOUT Clusters” в списке слева. Если вы планируете сохранить продакшн-компьютер как часть системы, введите его IP-адрес в поле “Production PC Address”. В противном случае оставьте это поле пустым. Переключение между управлением продакшн-компьютером и кластером дисплей-компьютеров происходит через установку флажка “Control Production Computer” (управление продакшн-компьютером).

	Cluster Name	Production PC	Control Production PC
Users	WO2	10.0.1.31	NO
WATCHOUT Clusters			
Instagram Feeds			
Modbus I/O Modules			
TCP/UDP Devices			

Cluster Name *
WO2

Control Production Computer
Production PC Address
10.0.1.31

Cluster Members

WP2

WP1

Имя, которое вы присваиваете кластеру, должно соответствовать имени кластера (“Cluster Name”), указанному в диалоговом окне «Настройки» (Preferences) в программе WATCHOUT. Использование одинаковых имен кластера и имен дисплей-компьютеров (см. далее) позволяет Blocks найти кластер в сети без указания IP-адресов дисплей-компьютеров.

Красная точка рядом с именем кластера в списке становится зеленой при успешном соединении с кластером. Сервер Blocks автоматически повторяет попытки соединения, пока оно не будет успешно установлено.

Добавьте дисплей-компьютеры в кластер командой “Add Member” (добавить участника) в меню “Add”. Если вы управляете продакшн-компьютером, такая команда не является необходимой, так как всё

WATCHOUT Cluster Member

Display Computer Name *
WP1

Preferred Primary in Cluster

IP Address (optional)

CANCEL OK

управление в любом случае будет проходить через продакшн-компьютер. Но если вы позже захотите удалить продакшн-компьютер и напрямую управлять дисплей-компьютерами, вы должны добавить и их, чтобы поддержать такой режим работы. При добавлении дисплей-компьютеров вы должны дать им имена. Эти имена должны соответствовать именам, заданным для дисплей-компьютеров в WATCHOUT.

Можно указать IP-адрес для каждого дисплей-компьютера. Примените этот метод, если вы используете фиксированные IP-адреса для дисплей-компьютеров в шоу WATCHOUT. Можно назначить один дисплей-компьютер первичным (“primary”) в кластере, он действует как мастер внутри своего кластера. Обычно неважно, какой компьютер становится мастером, и в этом случае можно не устанавливать флажок на эту опцию.

Управление WATCHOUT

После добавления в Blocks вы можете управлять WATCHOUT посредством кнопок, движков и задач. Вы также можете использовать события WATCHOUT, например, достижение конца временной шкалы, для переключения задач.

Instagram Feeds (загрузка из Instagram)

Добавьте и настройте загрузку Instagram, что впоследствии может быть использовано в блоке [Instagram](#).

The screenshot shows the 'Blocks' interface with a grid of image thumbnails and a configuration sidebar for an 'INSTAGRAM' block. The sidebar includes settings for 'Main/My Pix', 'Width' (1920), 'Height' (1080), 'Custom CSS URL', 'Custom HTML Classes', 'INSTAGRAM' feed settings, 'Feed' type (My Pix), 'Background Color' (Grey), 'Rows' (2), 'Columns' (4), and 'Skip images' (0).

Конфигурирование загрузки Instagram.

Вначале вы должны установить соединение с сервером Instagram, который предоставляет данные и изображения загрузки. Сервер Blocks имеет встроенную поддержку API Instagram, но необходимо получить действительные учётные данные для связи с сервером Instagram. Следуйте инструкциям [на сайте разработчика Instagram](#), чтобы получить такие учётные данные. Когда вы получили учётные данные, щёлкните на пустой квадратик в правом верхнем углу, чтобы ввести или изменить учётные данные.

▲ Базовые (sandbox) учётные данные позволяют отображать картинки только из собственного аккаунта. В частности, это не позволяет использовать хэш-теги. Обратитесь в Instagram для получения более привилегированных учётных данных.

Добавление загрузки (Feed)

Выберите “Instagram Feed” в меню Add, чтобы добавить загрузку. Канал имеет следующие параметры конфигурации.

Имя (Name)

Имя помещают в поле «Feed» при добавлении блока [Instagram](#).

Тип загрузки (Feed Type)

Определяет тип канала загрузки:

- **User (пользователь)**. Получайте изображения пользователя, указанного в поле “Tag/User Identifier”. Вы можете использовать имя “self”, как показано выше, для получения изображений, соответствующих вашей учётной записи.
- **Tag (тэг)**. Извлекает изображения, помеченные хэш-тегом, указанным в поле “Tag/User Identifier”.

Images to Cache (изображения в кэш-память)

Укажите количество изображений, которые необходимо сохранить на сервере Blocks. Изображения для блоков Instagram приходят из этого кэша, что позволяет показывать их, даже если сервер Instagram временно недоступен. Задайте некоторое их количество как минимальное для отображения из этой загрузки.

Active and Idle Refresh Intervals (интервалы активные и интервалы ожидания обновления)

Задается время между запросом серверу Instagram на новые изображения, при этом активное время используется во время просмотра блоков Instagram, а время ожидания используется в другое время. Следует установить время ожидания на большее значение, чтобы уменьшить риск превышения лимита скорости Instagram.

Модули I/O Modbus

Modbus – промышленный стандартный протокол для связи с такими устройствами, как датчики и реле. Blocks поддерживают версию протокола Modbus TCP, позволяя подключать такие устройства по обычной сети Ethernet.

Чтобы получить доступ к модулю ввода/вывода Modbus из Blocks, вы должны указать ему известный IP-адрес. Для правильного проведения процедуры обратитесь к документации производителя.

▲ Протокол Modbus TCP не имеет встроенной защиты. Не используйте модули ввода/вывода Modbus в открытой сети. Если вы хотите иметь общедоступную сеть Wi-Fi для функций мобильного гида, вы должны отделить эту сеть от сети управления так, чтобы гости могли получить доступ только к серверу Blocks. Например, используя правильно настроенный маршрутизатор и VLAN.

Blocks
MANAGE Edit Add

	Module	IP Address	Enabled	Notes	
Users	● MOXA	10.0.1.113	YES	Cabinet 2	Module Name * MOXA
Licenses					IP Address * 10.0.1.113
WATCHOUT Clusters					<input checked="" type="checkbox"/> Enable Module Note Cabinet 2
Instagram Feeds					
Modbus I/O Modules					Modbus Channel Number * 1
	Channel	Direction	Type	Alias	Alias (under *IO*) cleaningLights
	1	Out	Digital	cleaningLights	
TCP/UDP Devices	2	Out	Digital	curtain	
	3	Out	Digital	theaterDoor	
Art-Net/DMX Lighting	4	Out	Digital	DALI_5	Signal Direction <input checked="" type="radio"/> Output <input type="radio"/> Input
	10001	In	Digital	motion1	Signal Type <input checked="" type="radio"/> Digital <input type="radio"/> Analog
	10002	In	Digital	motion2	

Настройки модуля Modbus

Когда вы правильно сконфигурировали устройство Modbus с известным IP-адресом, добавьте его в список командой «Modbus Module» в меню «Add». Присвойте ему имя и укажите его IP-адрес. Затем указанное имя можно использовать для доступа и программирования каналов ввода-вывода (I/O).

Можно временно отключить модуль Modbus, сняв флажок “Enable Module” (включить модуль). В этом режиме никакие команды не будут отправляться ни на один из выходных каналов этого модуля.

Красный индикатор слева от имени модуля станет зелёным, когда сервер Blocks подключится к модулю.

▲ Протокол Modbus TCP предусматривает, что модули должны регулярно отключаться от сервера, при этом индикатор может временно становиться красным. Это нормально и не должно мешать работе устройства, так как сервер Blocks отсоединяется по мере необходимости.

Каналы Modbus

Большинство модулей ввода/вывода Modbus оснащены винтовыми клеммами для подключения контролируемых датчиков или устройств. Эти винтовые клеммы адресуются номером канала. Иногда номер канала указывается рядом с винтовой клеммой. Если нет, в документации модуля указывают соответствие между номерами каналов и винтовыми клеммами. Номера каналов обычно группируют в группы чисел, причём выходы начинаются с 1, а входы - с 10000, как показано в примере выше.

▲ Некоторые модули Modbus имеют настраиваемые номера каналов, тогда нужно будет также задать их при настройке модуля.

Чтобы добавить канал в модуль, выберите модуль в списке, затем выберите “I/O Channel” (канал ввода/вывода) в меню Add. Настройте параметры канала по мере необходимости.

Modbus Channel Number (номер канала Modbus)

Введите сюда номер канала, как указано в документации на модуль (или в соответствии с вашими настройками, если производится настройка номеров каналов).

Alias (псевдоним)

К каналу можно получить доступ по имени модуля Modbus (как «chN», где N - номер канала), но часто бывает полезно получить доступ к функции с помощью описательного имени. На приведенном выше рисунке каналу 1 присваивается псевдоним «cleanLights». Использование этого метода облегчает работу, поскольку вы можете просто переместить имя псевдонима, когда нужно переместить функцию на другой винтовой терминал или даже на другой модуль Modbus, и всё, что вы запрограммировали, продолжит работать.

Если вы дадите каналу псевдоним, то для программирования сможете получить доступ к этому каналу в виде «IO.YYY», где YYY - это псевдоним, присвоенный каналу.

Signal Direction (направление сигнала)

Указывает, является ли сигнал “Digital” (с режимами Вкл./Выкл., как например, реле) или “Analog” (например, датчик температуры).

Signal Type (тип сигнала)

Указывает, является ли сигнал “Digital” (с состояниями Вкл./Выкл., например, реле) или “Analog” (например, как датчик температуры).

Сетевые устройства

Многими устройствами можно управлять через локальную сеть с использованием стандартных IP-протоколов, таких как TCP или UDP. В некоторых случаях достаточно отправить устройству строку символов. Blocks могут управлять таким устройством, если добавить его как устройство TCP/UDP на странице Manage (управление) с указанием IP-адреса и номера порта. IP-адрес может быть назначен для устройства через меню конфигурации. Номер порта, как правило, фиксирован и указан в документации на устройство.

The screenshot shows the 'Blocks' interface with a 'MANAGE' section. A table lists devices, and a configuration panel is open for the 'Barco' device.

	Device	IP Address	Port	Driver	Enabled
Users	Barco	10.0.1.110	1025	BarcoPW392	Yes
WATCHOUT Clusters	Bowling	10.0.1.95	30471		Yes
Instagram Feeds	NEC2	10.0.1.31	4352	PJLink	No
Modbus I/O Modules					
TCP/UDP Devices					

Configuration panel for 'Barco':

- Device Name *: Barco
- Enabled
- IP Address *: 10.0.1.110
- Driver: BarcoPW392
- Port Number *: 1025
- Note:
- Communications Mode:
 - TCP
 - UDP

Чтобы добавить устройство, выберите “Network Device” в меню Add. Укажите его имя, IP-адрес и номер порта. Это имя будет использовано для программирования устройства, поэтому выберите короткое, но описательное имя. Если в документации не указано, следует ли использовать режим TCP или UDP, выберите TCP, поскольку он является наиболее распространенным.

В режиме TCP Blocks будет автоматически пытаться подключиться к устройству. При успешном соединении индикатор состояния в списке изменит свой цвет на зелёный. Режим UDP не поддерживает соединений и, следовательно, не имеет такого индикатора.

Драйверы сетевого устройства

Некоторые устройства имеют более сложные протоколы или требуют специального подтверждения или другого подхода к установлению надежного соединения. Надежно управлять такими устройствами может быть затруднительно, если просто отправлять на них командные строки. Кроме того, синтаксис, используемый некоторыми протоколами, не очень легко понимать и правильно применять.

Для преодоления подобных проблем можно использовать специализированные «драйверы» (“drivers”). Драйвер - это часть программного обеспечения, которая содержит подробные сведения о протоколе коммуникации и знает, как связаться с устройством надёжным способом. Драйверы устройств годятся для стандартных протоколов, такие обычно используются для видеопрокторов (например, [Barco pw392](#) и [PJ-LINK](#), используемые многими японскими производителями прокторов).

Дополнительным преимуществом драйвера устройства является то, что он может выставлять функции более высокого уровня, а не детали базового коммуникационного протокола. Например, вы нужно получить доступ к свойству «power» для включения или выключения проектора или, возможно, выбора

его входа для переключения видеовходов. Драйвер обычно предоставляет такие функции, позволяя связать эти свойства с кнопками и задачами для прямого управления.

Как только драйвер устройства будет установлен на сервере, его можно выбрать в меню “Driver” на панели настроек. Новые драйверы разрабатываются PIXILAB и партнёрами по мере необходимости. Свяжитесь с PIXILAB для получения самой свежей информации о доступных драйверах или ознакомьтесь со страницей [github](#) от PIXILAB, где мы храним большинство драйверов.

Свет: Art-Net / DMX-512

Театральным светом можно управлять с помощью протокола [DMX512](#). Некоторые другие системы освещения имеют «мосты» для управления по DMX512. DMX512 – простой протокол, позволяющий устанавливать или регулировать уровень для 512 каналов. Это может быть использовано для прямого управления интенсивностью большого количества источников света. Каналы можно комбинировать различными хитрыми способами для управления сложными осветительными приборами, такими как «движущаяся голова» с функцией панорамирования / наклона и другими эффектами и функциями.

DMX512 использует аппаратную проводку, применяемую для подключения источников света, однако данные DMX-512 можно передать и по проводу Ethernet, у которого есть и другие функции. Для «туннелирования» данных DMX через обычную сеть используют протокол [Art-Net](#).

К сети Ethernet подключают диммеры и другие устройства, напрямую совместимые со стандартом Art-Net. Кроме того, возможно соединение через адаптеры Ethernet-DMX, во множестве представленные на рынке.

Blocks могут управлять световыми приборами и диммерами, используя протокол Art-Net. Это происходит либо в интерактивном режиме – с помощью кнопок и ползунков, либо способом предварительного программирования Задач (Tasks). Для приборов, оснащенных разъемами DMX512 XLR, требуются адаптеры Ethernet-DMX, упомянутые выше.

Добавление прибора (Fixture)

Чтобы добавить прибор для управления, сделайте следующее:

1. Выберите “Art-Net/DMX lighting” на странице Manage, как показано ниже.
2. Нажмите кнопку + в правом верхнем углу.
3. Перетащите название модели из списка Model в основную область окна.

Blocks поставляется с набором описаний основных приборов (находится в разделе “Generic” Manufacturer). Описания других приборов находятся на странице <https://github.com/pixilab/blocks-fixtures>, вместе с инструкциями о том, как их установить, а также о том, как создать своё описание, если потребуется.

Blocks						MANAGE	Edit		+
Users	Name	Type	First Ch	Last Ch	Universe	Manufacturer			
Licenses	American XS	American DJ/XS 400	8	25	0	JB-Lighting			
WATCHOUT Clusters	one	Generic/Desk Channel	1	1	1	KAM			
Instagram Feeds	RGBA	Generic/RGBA Fader	4	7	0	MDG			
Modbus I/O Modules						Martin			
TCP/UDP Devices						Model			
Art-Net/DMX Lighting						Atomic 3000			
						MAC Aura			
						MAC Axiom Hybrid			
						MAC Viper AirFX			

Для базового одноканального управления выберите “Desk Channel” в разделе Model после выбора “Generic” (общий) в разделе Manufacturer (производитель). Чтобы добавить этот канал, перетащите “Desk Channel” в основную область окна. Несколько других базовых описаний можно найти в разделе «Generic», например, “RGBD Fader” для управления цветовой подсветкой с помощью каналов Red, Green Blue и Dimmer, или “Pan Tilt 16-bit” для управления движущейся головой с двумя каналами высокого разрешения для панорамирования и движения под наклоном.

Fixture Settings (настройки прибора)

После добавления описания прибора щелкните квадрат в верхнем правом углу, чтобы настроить его параметры, см. ниже. Введите имя, описывающее то, на что светит прибор, или какое-то другое осмысленное имя, поскольку оно будет использовано для программирования светового прибора. Укажите стартовый канал (номер канала DMX512), соответствующий настройке основного канала прибора. Если устройство использует более одного канала, дополнительные каналы отображаются последовательно после этого первого канала. Вы можете увидеть диапазон каналов, используемых прибором, в списке, в столбцах “First Ch” и “Last Ch”. Количество каналов, используемых прибором, регулируется его описанием .

Простое устройство “Generic Desk Channel” использует только один 8-битный канал, о чем свидетельствуют одинаковые цифры в столбцах “First Ch” и “Last Ch” (канал 1 в примере, показанном ниже). Устройство “Generic RGBA Fader” использует четыре канала с номерами от 4 до 7, причем канал 4 является базовым каналом.

The screenshot shows the 'Blocks' software interface. On the left, there is a sidebar with categories: Users, Licenses, WATCHOUT Clusters, Instagram Feeds, Modbus I/O Modules, TCP/UDP Devices, and Art-Net/DMX Lighting. The main area displays a table of fixtures. The 'one' fixture is selected and highlighted in blue. To the right of the table, the settings for the selected fixture are shown, including the fixture name, enabled status, start channel, artnet universe, and default fade rate.

	Name	Type	First Ch	Last Ch	Universe
	American XS	American DJ/XS 400	8	25	0
	one	Generic/Desk Channel	1	1	1
	RGBA	Generic/RGBA Fader	4	7	0

Fixture Name *
one

Enabled

Start Channel
1

Artnet Universe
1

Default Fade Rate
1

Artnet Universe

Art-Net / Ethernet имеет более высокую пропускную способность по сравнению с DMX512, поэтому можно туннелировать несколько шин DMX512 по одному проводу Ethernet. Шины DMX512 называются “universes” в протоколе Art-Net. Настройкой “Artnet Universe” вы указываете, какую Artnet Universe использовать для прибора, и она должна соответствовать настройке на принимающей стороне.

▲ Нумерация DMX каналов начинается с единицы (1), а нумерация вселенных начинается с 0 (нулевая вселенная). Некоторые устройства могут иметь другие настройки, что приводит к ошибкам “off-by-one”. И если ваше устройство не отвечает, вы можете попробовать сместить universe на один шаг выше / ниже числа, установленного в Blocks, и проверить, работает ли это.

Default Fade Rate (скорость перехода по умолчанию)

Этот параметр определяет, как быстро канал достигнет целевого значения при управлении с помощью кнопки. Управляя каналом из задачи, вы можете указать время затухания в каждом операторе задачи, не полагаясь на этот параметр.

Установите это значение на 0, чтобы достигнуть целевого значения не через постепенное затухание, а сразу при нажатии кнопки.

8. TASK (ЗАДАЧА)

Базовые функции управления могут быть выполнены привязкой кнопок и других элементов управления к контролируемым свойствам. Этот метод можно использовать для включения питания проектора при помощи кнопки или для задания уровня громкости при помощи движка. Однако в других случаях требуется одновременное управление несколькими функциями - например, включение нескольких проекторов и задание уровня освещённости в помещении - или выполнение команд в определённой последовательности. Такие более сложные сценарии управления могут быть выполнены с использованием задач.

The screenshot displays the 'Blocks' application interface for configuring a task. The top bar shows 'TASK Edit Realm Main'. The main area is divided into three sections: 'Group', 'Task', and 'Realm Variable'.

- Group:** Contains a 'Misc' group.
- Task:** Contains a 'StartShow' task, which is selected and highlighted in blue. Below it is 'TellBowling'.
- Realm Variable:** Contains a 'TheaterShow' variable with a value of 'false'.

The 'StartShow' task configuration is shown in a detailed view on the right. The trigger condition is 'now.getHours() < 17'. The task steps are:

- 1 **do** Network.Barco.power
set to TheaterShow.value
- 2 **wait** 1 seconds
- 3 **do** Network.Barco2.power
set to TheaterShow.value
- 4 **if** TheaterShow.value
- 5 **do** WATCHOUT.WO2.wakeUp
timeout? Literal value or expression
- 6 **do** WATCHOUT.WO2.load
showName "New Wine"
- 7 **do** WATCHOUT.WO2.play
timeline Literal value or expression
- 8 **end if**
- 9 **do** IO.DALI_5.value
set to !TheaterShow.value

The configuration panel on the right for 'StartShow' includes:

- Enabled:**
- Trigger:** Property Change
- Property to Watch:** Custom Path (advanced) Realm.Main.variable.Theater Show.value
- Property:** Realm Main variable TheaterShow value
- Current variable value:** Start when property Changes

Задача, которая включает или выключает проекторы и запускает презентацию.

Сервер Blocks может обрабатывать несколько задач. Задачи организованы в области задач (Realms), а затем - в группы (Groups). Задачи могут связываться со спотами (Spots) и другими устройствами, добавленными на странице Manage (такими как модули ввода/вывода I/O и сетевые устройства). Задачи могут также общаться друг с другом, вызывая друг друга или публикуя заметки, известные как переменные области *Realm Variables*, в «почтовых ящиках» (mailboxes).

▲ Задачи справляются с большинством сценариев управления, но Blocks поддерживает и расширенные сценарии, основанные на более формальном языке программирования. См. [«Дополнительные скрипты и драйверы на TypeScript»](#) для получения информации об этой опции.

Realm (область)

На верхнем уровне Задачи организованы в *области (realms)*. Как правило, вам нужна только одна область. Используйте несколько областей в более крупных системах, где нужно управлять несколькими более или менее независимыми областями, например, разными комнатами в здании. В каждой комнате может быть много вещей, которые можно контролировать - часто во взаимосвязанном режиме, а отдельные комнаты могут иметь мало общего или вообще не иметь ничего общего друг с другом. В этом случае имеет смысл управлять каждой комнатой как отдельной областью (realm).

Выберите текущую область, используя меню Область (Realm) в строке меню. Это меню также используется для добавления и удаления областей. Когда вы выбираете область в этом меню, списки Группа (Group), Задача (Task) и Переменная (Variable) изменяются для отображения этих объектов в данной области.

Переменные области (Realm Variables)

В дополнение к группам и содержащимся в них задачам область может также содержать общие переменные. Эти переменные напрямую доступны из всех задач в этой области. Они могут быть использованы для передачи состояния системы между задачами. Они также могут быть связаны с кнопками и движками, обеспечивая интерактивное управление этими состояниями системы.

Чтобы добавить Переменную области (Realm Variable), щёлкните на знак плюс в заголовке списка. Присвойте переменной имя и укажите её тип. Имя переменной должно начинаться с буквы и содержать только буквы, цифры и символ подчеркивания. Имя не должно содержать пробелов.

Выберите имя со смыслом, указав, для чего предназначена эта переменная. Это имя будет использовано для доступа к переменной из задач, а также для подключения к кнопкам и другим элементам панели управления.

Типы переменных

Переменная имеет фиксированный тип, определяющий, какие данные могут быть сохранены в ней, а также как она взаимодействует с кнопками и другими элементами управления.

- **Number (число).** Содержит числовое значение, которое может иметь дробную часть и может быть *отрицательным*.

- **String (строка).** Содержит произвольные символы. Строки буквенных символов (литеральные строки) должны быть заключены в одинарные или двойные кавычки.
- **Boolean (логические).** Значение является либо истинным (*true*), либо ложным (*false*).

После создания переменной можно настраивать эту переменную в панели настроек. Значение по умолчанию переменная будет принимать при запуске системы. Для числовой переменной вы можете указать её минимальное и максимальное значение, а также может ли переменная содержать дроби или должна ли быть целым числом. Эти ограничения будут применены при настройке переменной в панели управления или через задачу. Если вы зададите нулевое минимальное и максимальное значение, то никаких ограничений диапазона не будет.

Группа (Group)

Задачи организуют в группы. Добавьте группу, щёлкнув на знак «плюс» в заголовке группы. Создание групп задач имеет две цели:

- *Организационную, позволяющий объединить связанные задачи.*
- *Функциональную, для взаимоисключения задач внутри группы.*

Выберите группу, чтобы изменить её настройки, (см. справа). Чтобы отключить группу, снимите флажок “Enabled” (включено). Отключенные группы будут выделены курсивом. Их задачи не будут выполняться.

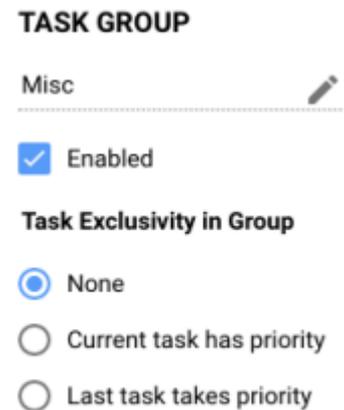
Эксклюзивность задач

Несколько задач могут выполняться одновременно, но это не всегда то, что вам нужно. Бывают группы взаимоисключающих задач, из которых только одна может выполняться в любой момент времени. Например, у вас могут быть задачи, которые подготавливают помещение к различным режимам презентации. В одном режиме проекционный экран должен быть развёрнут, а в другом он должен быть скрыт. Может потребоваться некоторое время для переключения режимов, чтобы экраны успели развернуться, а проекторы вышли из режима ожидания. Можно учитывать эти задержки с помощью операторов ожидания “wait” или предупредить задержки, вызванные некоторыми длительными действиями, отдавая команды в правильном порядке и в соответствующее время.

Пока потенциально продолжительная задача выполняется, вы можете изменить свое решение и нажмёте другую кнопку, чтобы выбрать другой режим, в этом случае всё может выйти из-под контроля, так как теперь у вас есть *две* задачи, которые будут конкурировать друг с другом, пытаясь установить свой режим в помещении.

Эксклюзивность задач может помочь избежать таких хаотичных ситуаций, гарантируя, что в любой момент времени может выполняться только *одна* задача в рамках группы задач. Это делается путем выбора одного из эксклюзивных режимов:

- **Current task has priority (текущая задача приоритетна).** Новая задача в этой группе не может быть запущена, пока выполняется другая задача. Текущая задача должна быть выполнена первой до её завершения.
- **Last task takes priority (последняя задача приоритетна).** При успешном запуске задачи в этой группе все уже выполняющиеся задачи в группе будут прерваны.



▲ Режим “Last takes priority” может иногда иметь неожиданные последствия, так как любая предыдущая задача может быть прервана на полпути. Вы должны убедиться, что задачи в такой группе задают все соответствующие состояния системы недвусмысленно, ясно, не ссылаясь на другие задачи.

Task (задача)

Задача – это список операторов с переключателем (триггером). Переключатель управляет выполнением задачи, а операторы внутри задачи определяют, что делать. Как вы видите на рисунке в начале этой главы, операторы занимают основную область, а переключатель отображается справа, когда задача выбрана в списке задач.

Узнайте больше о программировании задач на Blocks wiki: <https://int.pixilab.se/docs/blocks/tasks>

Создание и наименование задач

Добавьте задачу, щёлкнув на символ плюса в заголовке списка задач. Присвойте задаче описательное имя. Можно использовать любые символы в имени задачи, но иногда программирование задач становится проще, если вы начинаете имя с буквенного символа, а затем используете только символы, цифры и подчеркивания.

Enabled (включено)

Для задачи можно устанавливать флажок Enabled. Это позволяет и отключать задачи. Обратите внимание, что это дополнение к подобному флажку охватывающей группы. Для выполнения задачи должны быть установлены оба флажка. Отключенные задачи отображаются курсивом в списке задач.

Abort on Error (прервать при ошибке)

Если этот флажок установлен, любая ошибка, возникающая при выполнении задачи, приведет к немедленному завершению задачи, никакие следующие операторы не будут выполнены. Будет зарегистрировано сообщение об ошибке с указанием, что пошло не так. Это поведение по умолчанию. Иногда нужно, чтобы задача продолжала выполнять следующие операторы, даже если происходит ошибка. Если такой вариант вас устраивает, снимите этот флажок. Ошибка будет зафиксирована, но это не прервет задачу.

Переключатели (Triggers)

Существует несколько способов запуска задачи:

- Вручную, при нажатии на маленький треугольник рядом с названием задачи.
- При изменении состояния свойства системы. Это может быть начало воспроизведения на споте, обнаружение движения датчиком или изменение уровня громкости.
- По времени суток, что позволяет запланировать выполнение задач в нужный момент или несколько раз с некоторым интервалом.
- Запуск сервера, инициируется один раз при запуске сервера.

Явный запуск задачи

В дополнение к этим способам можно запустить задачу, напрямую связав кнопку на панели с «запущенным» (running) состоянием задачи или из другой задачи, используя оператор **do** (который будет описан далее в этой главе). Можно использовать этот явный метод независимо от режима переключения (Trigger) задачи. Установите для режима переключения значение “None” (нет), если вы хотите, чтобы задача запускалась именно таким образом.

Trigger	Property Change
Property to Watch	
Custom Path (advanced)	IO.motion1.value
Property	
IO	
motion1	
value	
Input	
Start when property	Becomes true

Переключатель изменения свойств (Property Change Trigger)

Чтобы переключать задачу при изменении какого-либо состояния системы, сначала выберите “Property Change” (изменение свойства) в меню Trigger, а затем укажите анализируемое свойство (property to watch), используя для этого поля, расположенные ниже. В показанном справа примере используются показания датчика движения, подключенного через входной канал модуля ввода-вывода (I/O) Modbus в качестве триггера.

▲ Для присвоения имени «motion1» была использована функция псевдонима ([Alias](#)) канала Modbus. Это делает программирование проще, чем при использовании имени, заданного по умолчанию по номеру канала Modbus (например, “ch10004”).

После выбора свойства, используемого в качестве переключателя, вы можете квалифицировать переключатель, используя опцию Start when property (начать, когда свойство...):

- **Becomes true (становится истинным)**, переключает задачу, когда датчик движения регистрирует движение (т.е. выходной сигнал датчика перекрыт).
- **Becomes false (становится ложным)**, переключает задачу, когда датчик больше не регистрирует движение (т.е. выходной сигнал датчика свободен).
- **Changes (изменения)**, переключает задачу при изменении значения независимо от направления. Это особенно полезно с числовыми или строковыми значениями и может быть дополнительно квалифицировано с использованием «Условие переключения» (Trigger Condition), что будет описано далее. Если для числового свойства выбрано “Becomes true” (становится истинным) или “Becomes false” (становится ложным), значение 0 считается ложным, а любое другое значение считается истинным. При подключении к строковому свойству пустая строка считается ложной, а непустая строка считается истинной (это применимо, даже если она содержит один пробел).

Time of Day Trigger (переключатель по времени)

Чтобы запустить задачу в определённое время, сначала выберите “Time of Day” в меню переключения, затем укажите время начала и другие ограничения, используя поля, расположенные ниже.

В первом поле Time задают время начала первого запуска задачи. Время следует вводить в 24-часовом формате (часы и минуты).

Затем вы можете указать дату начала; рекомендуем заранее создать задачу, чтобы можно было назначить дату её начала последующим числом. Аналогичным образом можно указать дату окончания.

▲ Дата окончания (Ending date) указывает последний день, когда будет запущена задача. Чтобы запустить задачу только на один день, задайте эту дату для даты начала и окончания выполнения задачи.

Задача, запускаемая на сутки, может выполняться один или несколько раз в течение определённого времени. Вы задаёте это, используя меню “Repeat” (повторить):

- **None** запускает задачу только один раз. Это полезно для режимов работы, которые выполняются только утром или вечером, например, выключение питания.

Trigger	Time of Day
Time	9:00
<input checked="" type="checkbox"/> Starting at Date	2017-10-26
<input checked="" type="checkbox"/> Ending after Date	2017-10-26
Repeat	At Interval
Interval	0:30
<input checked="" type="checkbox"/> Ending at Time	End Time 18:00

- **At Interval** задача перезапускается в заданном интервале относительно времени начала. Например, если время начала составляет 9:00, а интервал составляет 0:30, он также будет запущен в 9:30, 10:00, 10:30 и т.д. до достижения времени окончания (End Time).

- **Continuously (непрерывно)** Задача будет выполняться многократно, в течение дня, до заданного времени окончания (End Time). Используйте оператор ожидания (**wait**) в задаче, чтобы уменьшить частоту повторения.

Когда вы задаёте повторяющуюся задачу, вы можете указать время её окончания, после которого в этот день задача не будет выполняться повторно.

Trigger Condition (условие переключения)

Вы можете применить дополнительные ограничения в дополнение к настройкам триггера (Trigger). Например, при использовании режима “Changes” триггера Property Change вы можете принимать во внимание текущее значение, чтобы запускать задачу только при превышении определённого уровня. Или при использовании триггера Time of Day вы можете ограничить задачу в определённые дни недели.

Поле “Trigger Condition” над списком операторов может использоваться для применения таких ограничений. Синтаксис, используемый в этом поле, основан на JavaScript, поэтому ознакомление с этим языком программирования поможет вам максимально эффективно использовать эту функцию. Выражение, которое вы вводите сюда, имеет неявный доступ к переменной с именем *trigger*. В зависимости от режима запуска задачи эта переменная содержит одно из следующих значений:

- Если режим запуска – “Time of Day”; то это – объект JavaScript [Date](#). Этот объект предоставляет методы для определения текущего времени, даты, дня недели и т.д.

- Если режим запуска – “Property Change”; то это – число, логическое значение или строка, содержащая новое значение свойства.

Например, если режим запуска – “Property Change”, а наблюдаемое значение – уровень громкости, выраженный как дробное число от 0 до 1, то следующее условие запускает задачу только в том случае, если текущее значение выше 0,6:

trigger > 0.6

Если режим запуска – “Time of Day”, вы можете записать условие, подобное этому, чтобы запустить задачу во все дни, кроме воскресенья:

trigger.getDay() != 0

При этом запускается процедура `getDay` для объекта `Date`, чтобы ввести день недели числом 0 ... 6 (с воскресенья по субботу). После этого используется оператор «не равно» ([not equal to](#)) для сравнения введенного значения с числом 0 (которое соответствует воскресенью). Задача запускается только в том случае, если условие оценивается как истинное (`true`), которое будет выполняться в течение всех дней недели, кроме воскресенья.

Status, Errors and Logging (статус, ошибки, журнал)

Запущенная задача отображается номером справа от имени в списке задач (Task list). Этот номер указывает текущий номер строки задачи и может быть использован для отслеживания хода выполнения задач. Этот номер отображается только во время выполнения задачи. Для небольших задач этот статус иногда трудно увидеть, так как задача может выполняться в течение секунды. Вы можете замедлить выполнение задачи, вставив операторы **ожидания (wait)** в нужные позиции. Кроме того, вы можете использовать операторы журналов (**log**) для явного отображения текста и значений из задач. Такие

сообщения журнала отображаются в панели журнала, автоматически появляясь под операторами, когда это необходимо. Здесь также будут перечислены ошибки, возникающие при выполнении задачи.

Операторы

Операторы определяют, что и как будет делать задача. Операторы отображаются в центре страницы. На рисунке ниже показана страница Task при выборе задачи из списка Task list. Чтобы добавить оператор, щёлкните на знак «плюс» в верхнем правом углу и перетащите оператор из правого столбца в список.

Перетащите новые операторы из правой колонки в список операторов.

Изменить порядок следования операторов можно простым перетаскиванием. Для перемещения операторов между задачами сначала выберите операторы, а затем примените команды Cut/Copy/Paste в меню Edit. Выберите несколько операторов нажатием клавиш Shift-клик или Command/Control-клик.

Настройки операторов

Настройки оператора можно редактировать непосредственно в теле оператора. В качестве альтернативы после выбора оператора можно делать некоторые настройки в меню панели настроек справа.

Параметры оператора

Некоторые операторы очень просты, например, оператор ожидания, который принимает только одно значение. Другие более сложны, имеют несколько значений, называемых *параметрами* (*parameters*). Они используют выражения, похожие на те, что вы можете использовать в поле Trigger Condition (условие переключения), также основанные на синтаксисе JavaScript. Знакомство с языком JavaScript поможет вам при программировании задач.

Параметры иногда имеют обязательный тип. Например, единственный параметр для оператора ожидания (*wait*) должен быть числом, выраженным в секундах (определяет, сколько времени ждать). Чтобы указать значение для параметра, используйте либо точное значение, либо выражение.



*Оператор ожидания **wait** использует точные числовые значения.*

В некоторых случаях значение неизвестно заранее, но должно быть рассчитано во время выполнения задачи. Это делается с использованием *выражения* (*expression*), которое сходно с формулами, которые вы используете в электронной таблице. Например, вы можете указать значение в приведенном выше примере следующим образом:

1+1

Для вычисления значения используется *оператор сложения* (*addition operator*). Но поскольку это выражение использует только литеральные константы, у него нет никакого преимущества перед простым вводом числа 2. Чтобы создать что-то более интересное, нужно вставить в выражение некоторые *переменные*. Например, предположим, что вы хотите, чтобы задача подошла к этой точке в течение 4 секунд до полудня и через 2 секунды после полудня. Это может быть реализовано с помощью следующего выражения:

now.getHours() >= 12 ? 2 : 4

Это выражение сначала получает часовую составляющую текущего времени, посредством “now.getHours()”. Слово “now” (сейчас) - это глобальная переменная, которая всегда представляет текущее время, как объект [Date](#). Метод getHours () извлекает компоненту часов как число 0 ... 23. Затем это сравнивается со значением 12. Если оно больше или равно 12 (оператор «> =»), оно даёт значение 2, в противном случае оно даёт число 4. На этом последнем этапе используется [оператор JavaScript](#), который выбирает одно из двух этих значений в зависимости от результата логического (истинного / ложного) выражения. Подробнее об операторах JavaScript читайте здесь:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators

Что вы можете использовать в выражениях

Как было показано в примере выше, слово *сейчас* (*now*) можно использовать для установки текущего времени, как объекта [Date](#). *Math* и *Number* предоставляют некоторые полезные математические постоянные и функции. Литеральные (постоянные) значения указываются одним из следующих способов, в зависимости от их типа:

- 23 или -345.22 для чисел.
- *true* или *false* для логических операторов.
- “Will” или 'Smith' для строковых (текстовых) литералов. Вы можете использовать одиночные ('single') или двойные ("double") кавычки для заключения строки. Это должны быть 'прямые' кавычки, а не «кудрявые».

▲ Если вы новичок в программировании, то можете легко забыть проставить кавычки вокруг строковых литералов. Если вы всё-таки забыли и ввели Will вместо "Will", всё будет работать не так, как надо. Всякий раз, когда вы сомневаетесь, используйте оператор [log](#), чтобы узнать, какой результат выдаёт выражение.

Вы можете ссылаться на *локальную переменную* (*local variable*), просто указав имя локальной переменной, см. “[set local variable](#)” (задать локальную переменную) далее в этой главе. Можно также ссылаться на значение *переменной области* (*Realm Variable*):

NumVar.value

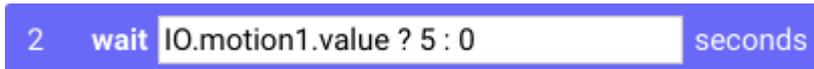
где NumVar - это имя переменной, см. «[Свойства и функции](#)».

▲ Вы должны добавить суффикс «.value» при доступе к значению реальной переменной (*real variable*), как показано выше. Это не относится к локальным переменным.

Наконец, у вас есть прямой доступ к статусу всей системы, это представлено *свойствами объекта* (*object properties*). Здесь используется тот же синтаксис, что и в поле “Custom Path” (пользовательский путь) триггера задач “Property Change”. В примере, показанном ранее (см. “[Property Change Trigger](#)”), это:

IO.motion1.value

Таким образом, ссылаясь на это свойство объекта, вы можете заставить оператор ожидания (*wait*) задерживать дальнейшее выполнение на 5 секунд, только если активирован датчик с именем «motion1»:



Те же правила и возможности применимы в поле *Trigger Condition* (условие переключения), где можно использовать те же постоянные, переменные, свойства объекта и операторы, а также доступна *триггерная переменная*, которая работает при изменении свойств или времени суток (она содержит новое значение свойства или текущее время как объект Date).

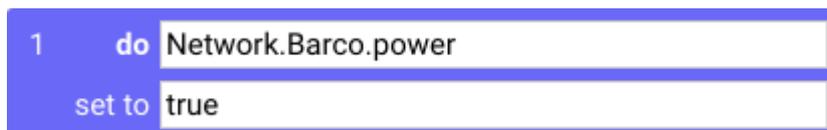
Подробнее о выражениях на Blocks wiki: <https://int.pixilab.se/docs/blocks/tasks>

do (выполнить)

Это наиболее часто используемый оператор. Как следует из его названия, это тот оператор, который вы используете для достижения цели. Он имеет две формы:

- *Setting a property* (*настройка свойства*) на системном объекте. В этом случае используется тот же синтаксис, что и при назначении кнопки для свойства, но при необходимости вы можете указать значение при помощи сложного выражения.
- *Calling a function* (*вызов функции*) на системном объекте. Такие функции предоставляют дополнительные возможности, недоступные, если просто задавать свойство.

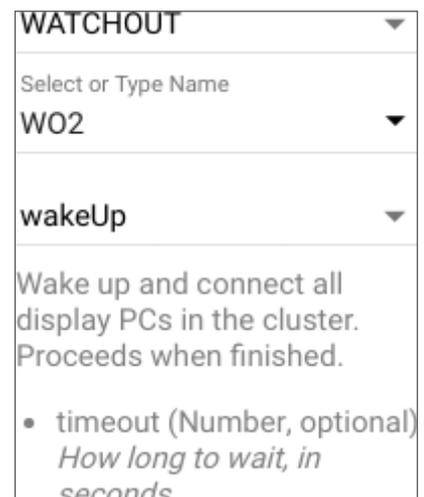
Вот пример для включения проектора «Barco», который управляется как сетевое устройство.



Используйте настройки Target на панели настроек, чтобы выбрать свойство, которое вы хотите задать, так же, как и при подключении кнопки. Если нужно, чтобы проектор *всегда* включался при запуске этого оператора, просто введите *true* (истина) в поле «set to». Поскольку это буквенное значение, не заключайте в кавычки *true* (см. «[Что вы можете использовать в выражениях](#)»).

Вызов функции

Не все функции управления имеют тип простого управления электропитанием или громкостью, где значение свойства непосредственно представляет эту функцию. Например, можно включить компьютеры в кластере WATCHOUT, используя функцию *wakeUp* (*просыпайся*). Вывод кластера из режима ожидания – это то, что должно произойти перед загрузкой и воспроизведением шоу, например, если его электропитание было отключено ночью.



5	do	WATCHOUT.WO2.wakeUp
	timeout?	Literal value or expression

Выполнение этого действия займет некоторое время. Вы не можете перейти к загрузке шоу до тех пор, пока система не будет запущена. Функция *wakeUp* делает это автоматически, заставляя задачу приостанавливать выполнение оператора до тех пор, пока включение не будет выполнено. Предоставляя контроль над временем ожидания, функция принимает параметр *timeout* и позволяет узнать, сколько времени придется ждать.

Этот параметр (*timeout*) является необязательным, как указано в его описании (и отмечено вопросительным знаком, следующим за его именем, как показано выше). Если вы не укажете значение, по умолчанию будет применено приемлемое значение.

Другой пример использования функции – это когда нужно принять во внимание несколько значений. Например, при управлении входом WATCHOUT (функция WATCHOUT часто используется для управления поведением WATCHOUT извне) вы можете указать как значение, так и скорость перехода для этого значения. Можно привязать ползунок непосредственно к входному значению, но ползунок использует только одно значение – значение, на которое установлен вход. Если вы хотите указать ещё и скорость перехода, следует использовать функцию *setInput* (*задать вход*). Эта функция может иметь до трёх параметров:

- Имя входа (строка).
- Значение для входа (число).
- Дополнительно: скорость перехода, в миллисекундах (число).

1	do	WATCHOUT.WO2.setInput
	name	"Arne"
	value	0.7
	rate?	8

Если вы не указали скорость перехода, она будет иметь такое же значение, как и при прямом вводе при помощи кнопки. Но если вы установите последний параметр на 1000 миллисекунд, то значение на входе (Input) будет изменяться постепенно, более 1 секунды, от текущего значения до достижения новой величины.

Использование выражений в операторах

Иногда нужно, чтобы оператор выполнял разные действия в зависимости от контекста. Например, задача, которая может включать или выключать электропитание в зависимости от состояния переменной или какого-либо другого состояния. Для этого используйте оператор *do*, но замените буквенное значение *true* (*истина*) выражением, как показано на рисунке в начале главы [Task](#). В этом примере мощность установлена на значение переменной области TheaterShow. Вы можете использовать любое допустимое выражение в поле “set to”, чтобы определить, включать или выключать электропитание.

Типы выражений и результаты

Свойства могут быть разных типов, это видно в нижней части панели настроек (см. рисунок справа). Это те типы, которые описаны в разделе

Network	▼
Barco	▼
power	▼
Power on/off (Boolean)	

«Типы переменных». Значение выражения, используемого для задания свойства, должно, как правило, иметь тот же тип, что и заданное свойство. Например, свойство «power» проектора – логического типа (Boolean), требует значений *true* (*истина*) или *false* (*ложь*). Уровень громкости требует цифрового выражения (обычно от 0 до 1). При использовании выражения результат иногда не совсем такой, какой вы ожидаете. Если операторы ведут себя плохо, можно вставить оператор **log** с тем же выражением, чтобы узнать, что происходит.

wait – оператор ожидания

Задерживает выполнение последующих операторов задачи на величину значения выражения.

```
2 wait 2 seconds
```

Вы можете использовать произвольное выражение, дающее положительное числовое значение (включая дробные части, например 0,5, чтобы подождать полсекунды).

log – оператор регистрации

Выводит результат выражения в окно журнала регистрации. Это окно появляется под списком операторов при выполнении оператора **log**, или когда что-то пойдет не так при выполнении задачи. Оператор журнала регистрации часто используется для определения значения выражения или локальной переменной и является весьма полезным средством для устранения неполадок, когда задачи не работают должным образом.

note (примечание)

Этот оператор не имеет никакого действия, он нужен для добавления заметок или комментариев к задаче. Например, с его помощью можно добавить описание того, что делает задача, или указать меры предосторожности, которые следует учитывать при изменении задачи.

Set realm variable (задать переменную области)

Устанавливает значение переменных области ([Realm Variables](#)) через выражение или в литеральном значении. Переменные области (Realm Variables) обычно используются для отслеживания состояния системы, естественно не представленного каким-либо уже существующим системным свойством. Они также полезны как средство для передачи информации между задачами. Кроме того, переменные области доступны как системное свойство для привязки к кнопке или ползунку, для переключателей задач (Task triggers), условий и выражений.

Для доступа извне к переменной области начинайте с уровня Realm, например:

```
Realm.Main.variable.TheaterShow.value
```

Переменная области в этом примере называется «TheaterShow» и существует в основной области (Main). Когда вы ссылаетесь на одну и ту же переменную Realm из задачи в основной области, вы можете просто ввести:

```
TheaterShow.value
```

Оператор set local variable (задать локальную переменную)

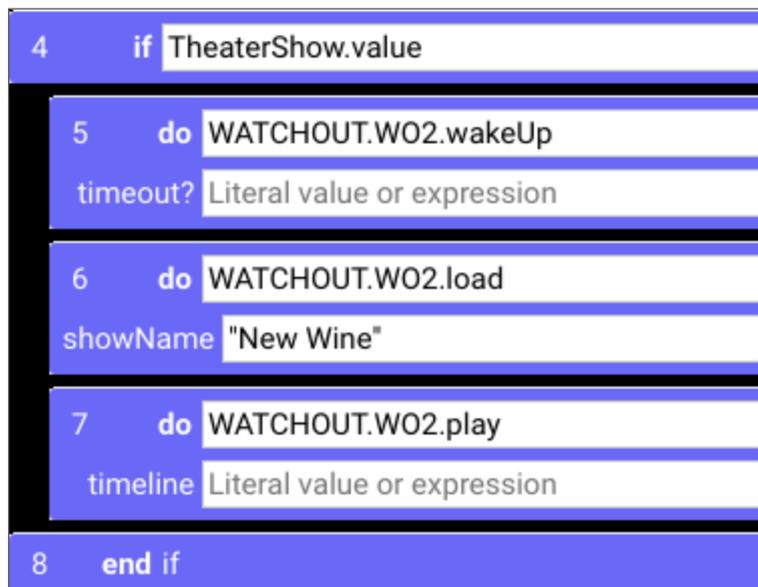
Локальная переменная существует только внутри задачи, где она определена. Используйте этот оператор для определения локальной переменной, а также для изменения её значения. Он не может быть доступен

вне пределов задачи и не может быть использован в переключателях задач или условиях. Применение локальной переменной – принять к сведению некоторое значение в какой-то момент задачи, а затем прочитать или изменить это значение в более поздней точке. Она также используется с операциями *if (если)* и *while (в то время, как)*, чтобы контролировать их поведение.

Локальная переменная не имеет заданного типа и других ограничений, как переменные области (Realm Variables). Локальная переменная принимает любое значение, которое вы в неё заложили, и может даже изменить тип, используя оператор *дать локальную переменную (set local variable)*. Кроме того, в отличие от переменных области, вы не можете так просто увидеть, какое значение имеет локальная переменная. Используйте оператор регистрации **log**, чтобы отобразить её текущее значение.

if (если)

Оператор *if (если)* позволяет назначать набор операторов, которые должны выполняться, только когда *если (if)* истинно. В примере, показанном ниже, утверждения 5, 6 и 7 будут выполняться только в том случае, если значение TheaterShow.value истинно. См. полный пример в начале этой главы, где эта переменная области (Realm Variable) управляет включением или выключением проекторов и запуском переключателя задачи. Переменной TheaterShow управляют напрямую кнопкой запуска и остановки театра. Только при истинном значении (true), кластер WATCHOUT должен выходить из режима ожидания, загружать и воспроизводить шоу, используя оператор *if (если)*.



Операторы if и end if выполняются только тогда, когда выражение в операторе if истинно.

Если результат выражения уже является логическим, как в приведенном выше примере, нет нужды записывать:

TheaterShow.value === true

чтобы убедиться, что это истина. Вместо этого просто используйте логическое значение как есть:

TheaterShow.value

Аналогично, если вы хотите выполнять операторы внутри только его значения ложь (*false*), не нужно писать

TheaterShow.value === false

Вместо этого припишите ссылку на значение с оператором *not* (prefix the value reference with the not operator), например:

!TheaterShow.value

Подробнее о выражениях и операторах JavaScript читайте здесь:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators

Оператор while

Оператор *while* очень похож на оператор **if**, но он выполняет внутренние инструкции *повторно* до тех пор, пока условие остаётся истинным. Примеры повторения некоторых операторов:

- Использовать один и тот же набор команд для нескольких целей.
- Повторить некоторую команду с предварительно заданным интервалом.

Можно, конечно, применять один и тот же набор команд к нескольким объектам, просто дублируя эти утверждения и изменяя целевой объект в каждой копии, но это быстро становится громоздким по мере увеличения количества целей. Если имена целей содержат последовательный номер, как показано в примере ниже, проще использовать цикл *while*, чтобы пройти через все цели. Пример, показанный ниже, относится к включению нескольких силовых реле в стойке. Имеется четыре реле, каждое из которых управляет электропитанием в отдельной части здания. Может потребоваться включать их последовательно, (а не сразу все) с небольшой задержкой между срабатыванием каждого реле.

```

1 set local variable newState to !!IO.power1.value
2 set local variable count to 1
3 while count <= 4
4   do IO["power" + count].value
   set to newState
5   set local variable count to count + 1
6   wait 0.5 seconds
7 end while

```

Силовым реле были даны последовательно пронумерованные псевдонимы: power1, power2, power3 и power4. Число в конце имени можно привязать к имени с помощью выражения конкатенации (сцепления) строки, как видно по выражению в квадратных скобках в операторе *do*. Значение локальной переменной *count* (отсчёт) сначала устанавливается равным 1 (в операторе 2). Конкатенация (сцепление) “power” со значением счётчика даёт «power1», это имя первого силового реле. Затем значение счётчика увеличивается на единицу оператором 5, а цикл *while* повторяет операторы, если значение меньше или равно 4 (номер последнего силового реле).

Красота этой конструкции по сравнению с простым дублированием операторов внутри цикла *while* заключается в том, что её можно легко изменить, если система расширяется, чтобы использовать шесть силовых реле вместо четырёх. Просто измените критерий в операторе *while* на:

count <= 6

а всё остальное оставьте без изменений. Аналогичным образом, если вы хотите увеличить задержку между включением каждого реле питания, достаточно изменить один оператор ожидания, вместо того, чтобы обновлять несколько операторов ожидания.

Комбинирование операторов *while* и *if*

Структурные операторы **if** и **while** могут быть использованы внутри друг друга. В приведенном выше примере задержка между переключением каждого реле электропитания действительно необходима только при их включении. Следовательно, вы можете исключить оператор **wait** при выключении, добавив оператор **if**, например так:

```

1 set local variable newState to !IO.power1.value
2 set local variable count to 1
3 while count <= 4
4   do IO["power" + count].value
   set to newState
5   set local variable count to count + 1
6   if newState
7     wait 0.5 seconds
8   end if
9 end while

```

Как и в предыдущем примере, локальная переменная *newState* сначала устанавливается в требуемое новое состояние (набором инверсии старого состояния первого силового реле, с помощью оператора *not*, отмеченного восклицательным знаком). Затем переменная *newState* используется для установки состояния силовых реле в операторе 4. Затем она также используется для управления оператором **if**, так что оператор **wait** внутри него будет применяться только тогда, когда значение оператора *newState* истинно (т.е. электропитание включено).

▲ Даже если не использовать операторы **wait**, выполнение всех операторов происходит за доли секунды. Следовательно, даже после вышеупомянутого изменения, реле не будут отключены абсолютно одновременно.

await (ждите)

Оператор `await` позволяет объединить несколько потенциально медленных операторов, а затем ждать их всех, а не по одному.

Некоторые операторы требуют времени для завершения. Самый очевидный пример – оператор `wait`. Но в некоторых случаях и оператор `do` заставляет себя ждать. Например:

- при загрузке шоу в кластер WATCHOUT;
- при включении проектора или дисплея.

Последующие операторы не будут выполняться, пока не сработает медленный оператор. Хотя выполнение задачи и ожидание завершения каждого оператора, как правило, является желательным поведением, это может помешать работе. Например, вы хотите включить пять дисплеев перед началом презентаций на этих дисплеях. Если дисплеи будут включаться по очереди, это вряд ли вас устроит. Лучше сделать так, чтобы дисплеи включались одновременно.

Это может быть выполнено с помощью оператора `await`. Проще говоря, медленные операторы выполняются параллельно между `await` и `end await`. Все они будут запущены сразу, а оператор `end await` отложит выполнение последующих операторов, пока медленные операторы не будут завершены.

Вы можете узнать, какие функции могут потребовать времени для выполнения, посмотрев описания системных объектов на странице [github PIXILAB](#). Любая функция, которая возвращает *Promise*, потенциально может занять много времени. Например, в объекте WATCHOUT вы можете видеть, что функции `connect` и `wakeUp` делают это. Таким образом, если оператор вызывается из задачи, задача может ждать, прежде чем двигаться дальше.

Аналогично, драйвер устройства может возвращать *Promise* в некоторых случаях, например, функция `powerUp` драйвера ChristiePerformance (он есть на github).

9. СВОЙСТВА И ФУНКЦИИ

В приложении Blocks свойства есть у многих объектов. Свойством мы будем называть полезное системное состояние, такое как включение или выключение электропитания или уровень громкости. Свойства могут быть напрямую выведены на элементы панели управления, например, кнопки, движки, индикаторы и текст. Они также могут быть назначены или использованы в более сложных сценариях управления, таких как описанные в предыдущей главе «Задачи» (Tasks).

Набор доступных свойств зависит от типа управляемого объекта. Например, дисплей-споты (Display Spots) могут быть включены или выключены через свойство электропитания (*power*), но это не относится к гостевому споту (Visitor Spot) по понятным причинам. В этой главе приводится краткое изложение доступных свойств для разных объектов управления.

▲ Некоторые свойства предоставляются через драйверы устройств ([device drivers](#)) и могут быть добавлены после публикации этого руководства. Рекомендуем всегда использовать информацию, предоставленную внутри Blocks, поскольку она будет самой оперативной.

Значения свойств являются двунаправленными (Bidirectional)

Важным преимуществом свойства объекта является то, что оно представлено *единственным значением* (*single value*), например, электропитание: его значение может быть либо истинным, либо ложным, или уровнем громкости, для которого задаётся определенный уровень, скажем, 0,5. Это позволяет легко управлять свойством – просто задать его значение. Этот механизм работает в двух направлениях; ползунок может использоваться для управления уровнем громкости, а также для указания его текущего значения. Если громкость изменяется по какой-либо причине, ползунок автоматически отображает новый уровень. Например, если Задача (Task) содержит оператор *do* для изменения уровня громкости, любой ползунок, прикрепленный к этому свойству уровня громкости, будет отображать новое значение.

Это позволяет значениям свойств действовать, как переключатели задач (Task triggers) в условиях переключения (Trigger Conditions) и в выражениях оператора. Любое свойство, представляющее полезное системное состояние (например, состояние электропитания – включено или отключено) может использоваться во всех этих случаях.

Свойства, доступные только для чтения (Read-only Properties)

Некоторые свойства нельзя задать напрямую, они просто отображают состояние устройства. Например, свойство спота *connected* указывает, подключен ли этот спот к серверу. Если нет, вы не сможете управлять этим спотом и, естественно, хотите знать об этом. Причиной может быть отсоединение сетевого кабеля, отключение электропитания и т.п.. Вы не можете заставить спот подключиться, задав *connected*, так как только сам спот может подключиться к серверу.

Свойства, заданные только для чтения, выделены курсивом в раскрывающихся меню выбора свойств. Они не могут быть использованы в качестве целей для операторов *do*, а также не могут управляться кнопками/движками. Но они *могут* использоваться как переключатели задач (Task triggers) или в выражениях.

Функции

Как было указано выше, свойство представляет собой *единственное значение* (*single value*). Оно может быть привязано непосредственно к элементу управления или индикатору. Однако в некоторых случаях требуется несколько значений. Например, чтобы задать цвет осветительного прибора, вы должны предоставить красный, зелёный и синий цветовые компоненты, а также скорость затухания (при необходимости). Такое поведение с несколькими значениями не может управляться с помощью свойства, у которого только одно значение. Вместо свойства должна быть использована функция. Функция может

быть использована только из задачи. Она не может быть напрямую связана с кнопкой. Чтобы запускать функцию «с кнопки»; вызовите функцию из задачи, затем подключите кнопку к свойству *пуска* (*running*) задачи, чтобы запустить её.

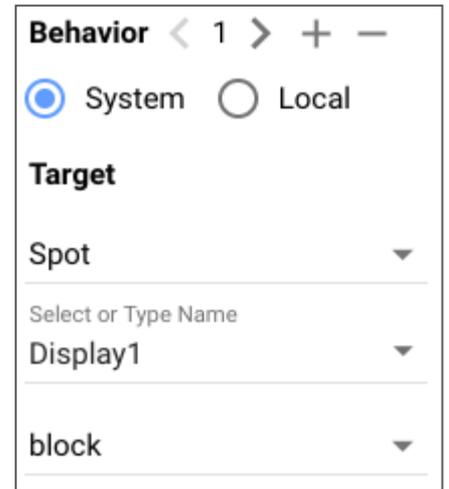
Параметризация путей свойств (Parametrization of Property Paths)

Путь к целевому свойству выбирается с помощью меню, как показано на иллюстрации справа. Получается фиксированный путь свойства, что означает, что конкретная кнопка предназначена для определённого свойства (а именно «Spot.Display1.block» в этом примере) независимо от его контекста.

Однако в некоторых случаях нужно, чтобы целевое свойство варьировалось в зависимости от контекста. Например, рассмотрим группу дисплей-спотов, каждый из которых управляется собственным iPad на подставке. Все дисплеи предоставляют доступ к одному и тому же контенту, и все iPad выглядят одинаково, поэтому на всех iPad должна быть одинаковая панель управления. Тем не менее, вы хотите, чтобы каждый iPad контролировал *свой* дисплей-спот. Этого нельзя сделать с помощью кнопок с фиксированными путями к свойствам, поскольку часть пути к свойствам (имя спота в данном примере) будет другой для каждой пары iPad / дисплей.

Одним из решений этой проблемы было бы дублирование блока, отображаемого на iPad, создание копий, соответствующих каждому целевому дисплею. А пути можно изменить, чтобы они соответствовали названию нужного целевого дисплея. Это работает, но приводит к решению, которое сложно поддерживать и обновлять новыми функциями для iPad.

Лучшее решение – использовать *параметр* (*parameter*) вместо жёстко заданного имени дисплея. Когда это возможно, в нижней части раскрывающегося меню появляется дополнительный пункт под названием [PARAMETER]. При выборе этого элемента открывается диалоговое окно, позволяющее вам определять параметры вместе с их *формальными значениями* (*formal values*).

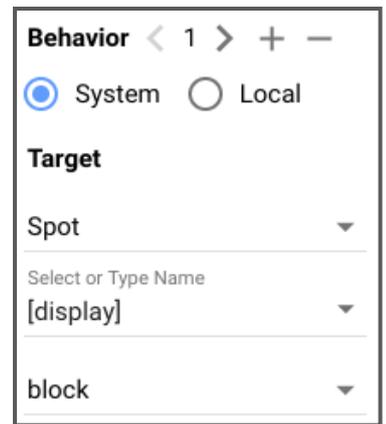


Добавьте параметр с именем «display», как показано выше. Задайте его *формальное значение* (*formal value*) на то, что можно использовать для проверки функциональности. Это формальное значение используется редактором для получения набора допустимых дочерних свойств, поэтому в этом примере оно должно быть именем репрезентативной области отображения. Кроме того, это значение будет использоваться в качестве значения по умолчанию для параметра, если не найден специфичный для спота параметр с таким же именем.

Чтобы использовать этот новый добавленный параметр вместо жёстко заданного имени дисплей-спота, выберите его, как показано выше, и нажмите кнопку SELECT. Это помещает [display] в раскрывающееся меню вместо фиксированного имени дисплея. Поскольку этот параметр разрешен для такого же имени (здесь Display1), всё по-прежнему работает так же.

Однако при размещении блока, содержащего эту кнопку, в споте, значение этого параметра можно переопределить для спота, используя аналогичный список параметров из настроек спота (см. «[Параметры](#)»). Таким образом, задав параметр *display* в имени целевого спота для каждого iPad, мы сделаем так, что каждый из iPad будет управлять своим дисплей-спотом, даже если все они используют один и тот же блок.

Такой же метод можно использовать для параметризации объекта любого типа, включая сетевые устройства, кластеры WATCHOUT, реле и т.д. Имейте в виду, что значение, назначенное параметру на споте, имеет приоритет над формальным значением, указанным для параметра с таким же именем в блоке. Вам необходимо убедиться, что полученное значение параметра имеет смысл для вашей системы, и обновить его, если вы переименуете объекты. Если объект не найден, кнопка не будет работать, и появится сообщение об ошибке, указывающее на эту проблему.



И наконец, вы можете использовать действие кнопки для динамического изменения значения параметра. Поскольку это локальное действие кнопки, оно изменяет значение параметра только для этого спота. Это дополнение к набору параметров спота, определяющих начальные значения каждого параметра спота. Если вы хотите изменить значение параметра, используйте кнопку с локальным действием, задаваемым с помощью настройки [Setting a Spot Parameter](#).

Дисплей-спот (Display Spot)

[Дисплей-спот](#) управляет фиксированными дисплеями, как правило подключенными по проводу Ethernet. Чтобы получить доступ к свойствам дисплей-спота (Display Spot), выберите «Spot» на верхнем уровне, затем имя спота или охватывающей группы спотов (Spot Group), и наконец, имя самого спота. Споты могут быть вложены внутрь группы спотов (Spot Group) на любую глубину, поэтому придется перемещаться при выборе свойства.

Property	Type	Описание
active	boolean	Спот активно просматривается. Например, блоком Locator.
block	string	Полное имя блока по умолчанию в виде “group/name”.
<i>connected</i>	boolean	Истина, если плеер подключен к серверу.
pictureSource	string	Имя альтернативного источника входного сигнала, например ‘HDMI’, если таковой поддерживается.
playing	boolean	Текущий блок воспроизводится (или временно остановлен). Указывает, что любой блок воспроизводится.
<i>playingBlock</i>	string	Имя основного воспроизводящегося блока, может появиться из расписания.
power	boolean	Истина, если электропитание включено. Не для всех спотов.
priorityBlock	string	Полное имя блока приоритета, в виде “group/name”. Переопределяет блок, заданный по умолчанию.
time	time	Позиция во времени при воспроизведении видео/звука. Необходим Синхронизатор (Synchronizer).
volume	number	Уровень громкости воспроизведения звука 0...1. Поддерживается не всеми спотами.

Свойства дисплей-спота. Свойства, доступные только для чтения, выделены курсивом.

В дополнение к указанным выше свойствам дисплей-спот также имеет функции, которые можно вызывать из задач:

Функция	Параметр	Описание
forceTags	tags	Через запятую укажите теги, которые требуется использовать принудительно (дополнительные теги могут быть указаны в настройках спота).
	(ofSet)	Набор тегов для воздействия (другие оставлены в покое).

gotoPage	path	Показать разделенный косой чертой путь блока внутри текущего корневого блока.
scrollTo	x	Прокрутите до указанной позиции (для работы нужен скроллер). Положение по X, 0... 1 или не определено.
	(y)	Позиция по вертикали, 0... 1
setCustomClasses	(value)	Класс или классы, разделенные пробелом. Удаляет все пользовательские классы, если они пусты.

Visitor Spot (гостевой спот)

[Гостевой спот \(Visitor Spot\)](#) обычно подключается через сеть Wi-Fi. Вы получаете доступ к его свойствам так же, как и в случае дисплей-спота ([Display Spot](#)), как было описано ранее.

Property	Type	Описание
block	string	Полное имя блока по умолчанию, в виде "group/name".
connected	boolean	Истина, если любое мобильное устройство подключено он-лайн.
playingBlock	string	Имя воспроизводящегося блока контента. Может появиться из расписания.
priorityBlock	string	Полное имя приоритетного блока, в виде "group/name". Переопределяет блок, заданный по умолчанию.

Location (местоположение)

Вы получаете доступ к свойствам [Location](#) так же, как и в случае дисплей-спота ([Display Spot](#)), описанного выше. Блоки, связанные с местоположением (Location), появятся на гостевых спотах при навигации к этому местоположению.

Property	Type	Описание
block	string	Полное имя блока по умолчанию, в виде "group/name".
playingBlock	string	Имя основного воспроизводящегося блока, может появиться из расписания.
priorityBlock	string	Полное имя приоритетного блока, в виде "group/name". Переопределяет блок, заданный по умолчанию.

Задача (Task)

Задачи (Tasks) и связанные с ними переменные области ([Realm Variables](#)) доступны в объекте область (Realm) следом за именем области. Затем для задачи выберите *группу* следом за именем группы и задачи. Или, для переменной области, выберите *переменную*, за которой следует имя переменной.

Свойство запуска (*running*) задачи истинно во время выполнения задачи. Установите для этого свойства значение «истина» (*true*), чтобы запустить задачу непосредственно от кнопки или другой задачи. Вы также можете установить значение «ложь» (*false*) для немедленного завершения задачи.

▲ Если задача имеет условие переключения (Trigger Condition), это условие будет оцениваться даже при использовании этого метода, и задача будет запущена *только* в том случае, если условие выполнено.

Для переменной области (Realm Variable) свойство *значение* (*value*) – это текущее значение, которое будет иметь тот же тип, что и переменная. Назначьте изменение этого значения от кнопки или другой задачи.

При программировании задачи в той же области используйте оператор “[set realm variable](#)” (задать переменную области), чтобы задать значение и прочитать его, используя только:

VarName.value

где *VarName* – это имя переменной области. Нужно использовать только полное имя (начиная с уровня корня области) при доступе к переменной области из элементов управления панели или задач вне этой области.

Канал Modbus

Устройства управления, добавленные в модули входа/выхода [Modbus I/O Modules](#) на странице Manage. Единственное свойство канала Modbus – это его значение. Если канал является входом, то значение будет доступно только для чтения. Если канал установлен на «Analog», то значение будет числом в диапазоне 0 ... 1, в противном случае оно будет логическим.

▲ Если вы присваиваете каналам Modbus псевдоним (Alias name), то можно получить доступ к этим каналам по имени (IO name). Это бывает полезным, когда нужно присвоить каналу не номер, а понятное имя, описывающее его функцию.

Art-Net/DMX

Чтобы обратиться к световому прибору, сконфигурированному, как описано в разделе «[Свет: Art-Net / DMX-512](#)», сначала выберите «Artnet», затем имя осветительного прибора, а затем требуемое свойство или функцию. Большинство каналов имеют свойство *value*, которое, как правило, стандартизировано до диапазона 0 ... 1, независимо от своего фактического разрешения. Диапазон обычно указан под выбранным свойством.

Если вы установите значение свойства, будет применена установленная для прибора скорость затухания по умолчанию ([Default Fade Rate](#)). Если же вместо этого вы выбрали функцию *fadeTo*, вы можете указать как целевое значение, так и скорость затухания.

Для некоторых источников света с распространёнными функциями, такими как цветовые каналы RGB или функциями панорамирования / наклона, может быть предусмотрен дополнительный параметр с именем *multi-Func*, позволяющий устанавливать несколько значений с помощью одного оператора задачи.

Некоторые свойства могут принимать дискретные значения, которые задаются строкой. Для свойств, которые имеют и дискретное состояние, и соответствующее числовое значение, сначала выберите дискретное состояние, затем (с помощью второго оператора **do**) установите соответствующее значение.

WATCHOUT

Управляет подсистемой WATCHOUT, добавленной в [WATCHOUT Clusters](#) на странице Manage.

Property	Type	Описание
<code>auxTimeline</code>	<code>timeline</code>	Для управления дополнительными шкалами (auxiliary timelines) шоу.
<code>input</code>	<code>number</code>	Текущее значение входа (как задано в Blocks).
<code>connected</code>	<code>boolean</code>	True, если кластер подсоединён, false, если кластер находится оффлайн.
<code>playing</code>	<code>boolean</code>	True, если воспроизводится основная временная шкала (main timeline).
<code>ready</code>	<code>boolean</code>	True , если кластер полностью действующий, false , если кластер busy (занят) (например, загружает файлы).
<code>showName</code>	<code>string</code>	Имя текущего загруженного шоу. Например, имя дисплей-компьютера или полный путь для продакшн-компьютера.
<code>standBy</code>	<code>boolean</code>	True , если кластер находится в режиме ожидания.
<code>time</code>	<code>time</code>	Текущее положение во времени на основной временной шкале. В задачах выражается в миллисекундах.

При выборе `auxTimeline` необходимо выбрать или ввести имя дополнительной временной шкалы, которая затем предоставляет доступ к следующим дополнительным свойствам (sub-properties):

Property	Type	Описание
<code>playing</code>	<code>boolean</code>	True , если дополнительная временная шкала воспроизводится, false , если она временно приостановлена или остановлена.
<code>stopped</code>	<code>boolean</code>	True, если дополнительная шкала остановлена, false, если она воспроизводится.
<code>time</code>	<code>time</code>	Текущее положение во времени на дополнительной временной шкале. В задачах выражается в миллисекундах.

В дополнение к вышеуказанным свойствам кластеры WATCHOUT предоставляют следующие функции, которые могут быть вызваны из задач:

Функция	Параметр	Описание
connect		Попытаться подключиться к кластеру. Продолжать после завершения.
disconnect		Отключиться от кластера.
gotoControlCue	name	Имя управляющего сигнала (control cue) для быстрого перехода.
	reverseOnly	Искать только назад во времени для указанного управляющего сигнала.
	timeline	Имя временной шкалы для управления (это основная шкала, если не указано другое).
gotoTime	time	Время перехода (указано в миллисекундах).
	timeline	Название временной шкалы для управления (это основная шкала, если не указано другое).
load	showName	Загрузка шоу с задержкой выполнения задачи до завершения загрузки. Указывает полный путь при управлении продакшн-компьютером, иначе просто показывает имя. Всегда будет загружаться, даже если это шоу уже загружено.
pause	(timeline)	Приостановка временной шкалы. Название временной шкалы, не указано для основной шкалы.
play	(timeline)	Воспроизведение временной шкалы. Параметр – имя временной шкалы, не указано для основной шкалы.
reset		Остановка всех временных шкал.
setInput	name	Установить значение названного входа.
	value	Значение для установки.
	(rate)	Скорость перехода, в мс.
setStandBy	standby	Установить состояние ожидания для кластера.
	(rate)	Скорость перехода, в миллисекундах.
stop	(timeline)	Остановить временную шкалу. Название временной шкалы, не указано для основной шкалы.
wakeUp	(timeout)	Проснуться и подключить все дисплей-компьютеры кластера. Продолжать после завершения.

Сеть (Network)

Элементы управления добавляются в «[Сетевые устройства](#)» на странице Manage. При использовании без драйвера устройства доступно свойство «только для чтения» (только TCP) и следующие функции:

Функция	Параметр	Описание
enableWakeOnLAN		Подготавливает это устройство к действию по команде wakeOnLAN (запоминает его MAC-адрес)
sendBytes	toSend	Отправить массив необработанных байтов данных.
sendText	toSend	Текст для отправки на удаленное устройство.
	(eoln)	Конец строки. По умолчанию – возврат каретки.
wakeOnLAN	(macAddress)	Отправьте wake-on-LAN на это устройство или на указанный MAC-адрес (выраженный в виде строки шестнадцатеричных байтов, разделенных двоеточиями, или в виде массива из 6 чисел).

Если есть драйвер устройства, он обычно предоставляет несколько свойств более высокого уровня (например, включение/выключение питания или выбор входа при управлении видеопроектором). Он показывает или не показывает вышеуказанные функции. В большинстве случаев драйвер также управляет соединением с устройством, пытаясь постоянно повторно подключаться, пока не будет установлено соединение.

10. АВТОРИЗАЦИЯ И РОЛИ

Blocks предоставляет фиксированный набор ролей, которые могут быть назначены пользователям (см. «[Users. Пользователи](#)»). Пользователю всегда предлагается войти в систему при использовании редактора Blocks. При доступе к споту вход в систему не обязателен и обычно не используется (см. «[Role Required to Use Spot](#)»).

Доступны следующие роли пользователя:

Роль	Описание
Admin	Полный контроль над всеми аспектами Blocks, включая пользователей и их роли.
Manager	Полный контроль над всеми аспектами Blocks, кроме пользователей и их ролей.
Creator	Может создавать и редактировать контент всех видов.
Editor	Может редактировать существующий контент.
Contributor	Может заменять контент в блоках верхнего уровня: текстовом и медиа.
Staff	Зарегистрированный пользователь, который не может ничего редактировать (может быть использовано для контроля доступа).

Каждая роль представляет собой подмножество вышестоящей роли. А каждая высшая роль обладает всеми возможностями низших ролей, плюс свои возможности. То же самое относится и к использованию этих

ролей для настройки ограничений доступа, как описано ниже. В нижеследующей таблице описаны разрешения редактора Blocks для каждой роли.

Функция	Admin	Manager	Creator	Editor	Contributor
Пользователь: добавить / удалить / настроить	√				
Управляемые объекты: добавить / настроить	√	√	√		
Управляемые объекты: удалить / переименовать	√	√			
Спот: добавить / настроить / переназначить	√	√	√		
Спот: удалить / переименовать	√	√			
Добавить блок / расписание	√	√	√		
Редактировать блок	√	√	√	√	
Переименовать блок	√	√	√		
Удалить блок	√	√			
Редактировать расписание	√	√	√	√	
Расписание: удалить / переименовать	√	√			
Редактировать / заменить контент корневого медиа / текстового блока	√	√	√	√	√
Задача / Группа / Переменная добавить / настроить / изменить	√	√			
Задача / Группа / Переменная удалить / переименовать	√	√			

Настраиваемые ограничения доступа

В дополнение к фиксированным ограничениям редактора, перечисленным выше, возможно также настроить специфичные для объекта ограничения для различных функций.

Функция	Предоставляется с помощью
Просмотр спота Display или Visitor	настройки спота “Role Required to Use” (роль, необходимая для использования)
Увидеть блок в редакторе	настройки блока или группы “Role Required to See” (роль, необходимая для просмотра)
Назначить (перетащить) споту блок или расписание	настройки спота или вмещающей группы “Role Required to Control” (роль, необходимая для контроля)

Как обсуждалось выше, роли образуют правильные подмножества. Следовательно, если вы выберете роль *Creator* в качестве ограничения для доступа, вы неявно также предоставите доступ пользователям *Manager* и *Admin*. спота.

Роль, необходимая для использования спота (Role Required to Use)

Вы можете ограничить доступ к дисплей-споту (Display) или гостевому споту (Visitor) с помощью ролей. Это делается через параметр “Role Required to Use Spot”, который находится на вкладке Advanced настроек спота.

The screenshot shows the 'Display' settings dialog with the 'ADVANCED' tab selected. The 'Role Required to Use Spot' dropdown menu is open, showing 'Staff' as the selected option. Below it, the 'Role Required to Control' dropdown is set to 'None'. At the bottom, there is an unchecked checkbox labeled 'Power Down when no Block shown'. The 'CANCEL' and 'OK' buttons are visible at the bottom right of the dialog.

Если вы выберете здесь какой-либо вариант, кроме «None» (нет), потребуется вход в систему, прежде чем спот отобразит свой контент. Следовательно, этот параметр полезен только для устройств отображения, у которых есть клавиатура для ввода учетных данных. Этот параметр предназначен для спотов дисплей (Display) или гость (Visitor), используемых как панели управления для системных объектов с ограничениями по использованию. Кнопка или другой элемент управления, прикрепленный к системным объектам с ограниченным доступом, не действуют, если спот не был зарегистрирован пользователем с соответствующей ролью настройками «Роль, необходимая для управления» (см. ниже).

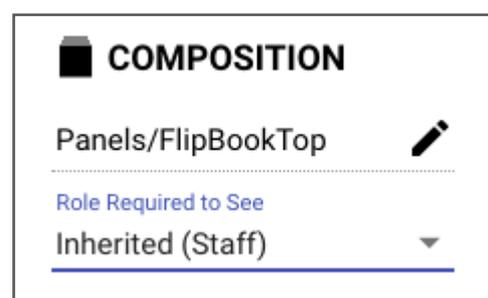
Роль для просмотра блока в редакторе (Role Required to see)

Вы можете ограничить круг пользователей, которым разрешён просмотр определённых блоков в Редакторе (вкладка **Edit**). Скрывая блоки от некоторых пользователей, вы уменьшаете вероятность ошибок, показывая только блоки, разрешённые для роли вошедшего в систему пользователя. Вы можете указать роль либо на уровне группы блоков, либо для каждого отдельного блока. Следовательно, если блоки, которые «запрещены» для некоторых ролей, уже сгруппированы вместе, вы можете просто установить ограничение для группы. Для этого сначала выберите группу с помощью строки меню в верхней части страницы «**Display**», а затем щёлкните на значок замка рядом с именем группы.



При этом открывается диалоговое окно, в котором можно указать минимальную роль, необходимую для просмотра блоков этой группы в редакторе. После установки такие блоки будут скрыты от пользователей с более низкой ролью.

Кроме того, вы можете задать минимальную роль, необходимую для просмотра блока в самом блоке. Эта настройка доступна только на самом верхнем уровне блока. По умолчанию блоки наследуют роль, указанную на уровне группы, как, например, указано параметром “**Inherited**” (унаследовано) на рисунке справа. Однако вы можете использовать настройку на уровне блока, чтобы переопределить унаследованный уровень группы, сделав его более ограничивающим или более разрешающим.



Role Required to Control (роль, необходимая для управления)

Блоки обеспечивают детальный контроль над ролями пользователей и их возможностями по управлению. Ограничения управления определяются для каждого объекта и представлены значками замков во всех блоках. Замки контролируют управление извне, исходящее от кнопок и других элементов управления, появляющихся на спотах. Изначально для роли, необходимой для управления всеми объектами, задано значение *None*, и это означает, что для управления объектами не требуется особая роль. Этот разрешающий параметр соответствует поведению предыдущих версий **Blocks** и гарантирует, что функции управления работают так же, как и раньше.

Ограничения управления задаются ролями, перечисленными выше, но существует ещё псевдороль, называемая «*Идентифицированный спот*» (*Identified Spot*). Эта псевдороль не требует входа в систему и предоставляется всем спотам **Display** и **Visitor** (гость) в системе, кроме спота **Visitor** по умолчанию, как назначено настройками [serverRootRedirect](#). Этот параметр обеспечивает некоторую безопасность, поскольку посетители, использующие имя спота **Visitor**, заданное по умолчанию, не могут управлять такими объектами. Только споты, идентифицирующие себя по **ID**, известному системе, или использующие явный **URL**-адрес спота **Visitor**, считаются идентифицированными спотами (*Identified Spot*). Следовательно, это полезная настройка по умолчанию для всех объектов, не предназначенных для управления случайными посетителями (или хакерами).

Все роли, кроме «*None*» и «*Identified Spot*», требуют, чтобы управлял объектом лишь зарегистрированный пользователь. Кроме того, вошедший в систему пользователь для управления объектом должен иметь указанную роль или более высокую роль. Поведение при входе в систему должно запускаться параметром “[Role Required to Use a Spot](#)”, описанным выше. Если пользователь с требуемой ролью не вошел в систему, кнопка или другой элемент управления не будут работать, и будет зарегистрировано сообщение об ошибке.

Конечно, вы не предоставите «опасные» функции управления в общедоступных спотах для посетителей, это явное ограничение обеспечивает дополнительную защиту от злонамеренных попыток контролировать функции системы. Оно действует как внутренний барьер, проверяя все попытки, поступающие извне, следя за тем, чтобы они подчинялись правилам доступа, определенным для каждого объекта.

▲ ПРИМЕЧАНИЕ. Ограничение “Role Required to Control” применяется только к попыткам управления извне (например, из спотов). Внутреннее управление в самих блоках (задачи и сценарии) не подпадает под эти ограничения.

Символ замка, указывающий на применяемые ограничения:



Ограничение управления не применяется.



Ограничение управления применяется.



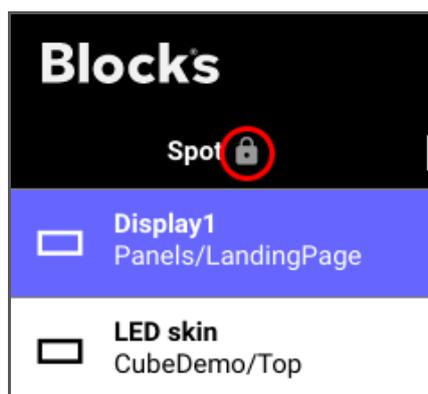
Применяется более строгое ограничение управления по сравнению с родительским объектом или группой.



Применяется менее строгое ограничение контроля по сравнению с родительским объектом или группой.

Споты

Роль, необходимая для управления (“Role Required to Control”), контролирует все свойства спотов, такие как *block*, *priorityBlock*, *power*. Роль может быть установлена в разделе ADVANCED в настройках спота. Если не переопределено, роль наследуется от вмещающей группы спотов (как указано в настройках группы) или на верхнем уровне – от общей настройки “Role Required to Control”, доступной через символ замка над перечнем спотов.



Управляемые объекты

Все управляемые объекты, найденные на странице Manage, имеют настройки ограничения управления, на что указывают символы замков. Настройки замка, отображаемые в списке управляемых объектов, определяют ограничения по умолчанию для каждой категории (кластеров WATCHOUT, модулей ввода / вывода Modbus, устройств TCP / UDP и световых приборов Art-Net / DMX).

При выборе категории отображаются ее объекты, и для каждого из этих объектов можно переопределить настройку, унаследованную от категории. Наконец, для некоторых объектов (таких как выходы Modbus) вы можете переопределить эту настройку для каждого отдельного выхода.

На иллюстрации, показанной ниже, ограничения применяются к кластерам WATCHOUT и к модулям ввода-вывода Modbus на уровне категории.

Модуль “Netio” Modbus предъявляет более строгие требования к роли, о чём свидетельствует знак «плюс» на символе замка. Все его выходные каналы имеют одинаковые требования, кроме канала 103, который является менее ограничивающим.

Чтобы увидеть требование к роли, заданное на каждом уровне, и изменить эту настройку, щёлкните на символ замка на нужном уровне.

Blocks						MANAGE	Edit	Add	
Users	● f	192.168.2.32	YES			Module Name * Netio			
	● g	192.168.2.34	YES						
Licenses	● MOXA	10.0.1.113	YES	Cabinet 2		IP Address * 192.168.2.24			
	● Netio	192.168.2.24	YES						
WATCHOUT Clusters									
Instagram Feeds	Channel	Direction	Type	Alias		Modbus Channel Number * 103			
	101	Out	Digital	Power1					
Modbus I/O Modules		102	Out	Digital	Power2		Alias (under 'IO') Power3		
		103	Out	Digital	Power3				
TCP/UDP Devices		104	Out	Digital	Power4		Signal Direction <input checked="" type="radio"/> Output <input type="radio"/> Input		
Art-Net/DMX Lighting							Signal Type <input checked="" type="radio"/> Digital <input type="radio"/> Analog		

Задачи и переменные области (Tasks and Realm Variables)

Аналогичная иерархия ограничений управления применяется к задачам и переменным области. Ограничение верхнего уровня здесь указывается на уровне области, как определяется символом замка рядом с названием области в верхней части окна. Этот параметр определяет ограничение, применяемое к переменным области и группам задач, если оно не переопределено. Все параметры, применяемые на уровне группы, наследуются задачами в этой группе, если они не переопределены конкретной задачей.

Для задач это ограничение определяет, кто может запускать задачу, устанавливая её свойство *запуска* (*running*) с помощью кнопки. Это не влияет на нормальный запуск задач, как указано в настройке *Триггер* *Trigger* для каждой задачи.

Аналогично, для переменных области определяется, кто может устанавливать значение переменной с помощью кнопки, движка или другого элемента управления. Это не влияет на способность задания переменных области изнутри (например, из операторов задачи).

▲ **ВАЖНО:** Поскольку задачи редко предназначены для непосредственного запуска анонимными посетителями, обычно рекомендуется устанавливать для роли уровня Области Realm более

строгую настройку, чем *None*. Впоследствии это может быть смягчено для отдельных задач или переменных, если потребуется.

А. ФАЙЛОВАЯ СТРУКТУРА СЕРВЕРА

Сервер Blocks хранит свои данные, а также ваши медиафайлы и параметры в папках. Для использования Blocks не нужно знать, как выглядит файловая структура, но если, например, вы хотите переместить что-либо с одного сервера Blocks на другой, это может пригодиться.

По умолчанию Blocks будет хранить свои данные в каталоге с именем “PIXILAB-Blocks-root” в домашнем каталоге учётной записи пользователя, под которой запускается сервер.

PIXILAN-root	Корневой каталог данных Blocks, расположенный в пользовательском каталоге
config.yml	Конфигурационный файл сервера
logs	Файлы журналов регистрации, иногда полезные при устранении ошибок сервера.
model	Здесь настройки страниц Manage, Task, Spots и Schedules.
Core	Внутренние данные сервера
Instagram	Информация о загрузках (feeds) Instagram, добавленных на странице Manage
ModbusModules	Содержит один файл на модуль Modbus, добавленный на стр. Manage.
NetworkDevices	Содержит один файл для каждого модуля сети, добавленного на стр. Manage.
Schedules	Содержит все блоки Schedule.
Main	Каталог, содержащий группу “Main” (по каталогу на группу).
Entrance.json	Файл, описывающий расписание “Entrance” (вход) для основной группы (“Main”).
Spot	Файл, описывающий конфигурацию всех спотов.
Task	Каталог областей задач (Task Realms), один файл для каждой области.
Users	Файл с информацией о пользователях, добавленных через страницу Manage.
WOClusters	Содержит один файл на кластер WATCHOUT, добавленный на стр. Manage.
public	Каталог, содержащий общедоступные файлы (http).
block	Все блоки, которые вы создаете (кроме расписаний Schedules), находятся здесь.
Main	Группа блоков с именем “Main”.
Saltkrakan	Блок с именем «Saltkrakan».
media	Медиафайлы, используемые в этом блоке, находятся в этой директории.
Spec.pixi	Данные, описывающие структуру блока.
script	
driver	Файлы драйверов для сетевых устройств хранятся здесь

Перенос конфигурации сервера

Вы можете передать целые группы блоков, а также отдельные блоки, скопировав блок или всю папку вмещающей группы с одного сервера Blocks на другой. Вам не нужно перезапускать сервер после добавления к нему новых блоков. Просто перезагрузите страницу Display в редакторе, чтобы отобразить новые блоки.

Аналогичным образом вы можете переносить расписания, сетевые устройства, модули Modbus, задачи и драйверы устройств, копируя эти каталоги или файлы с одного сервера Blocks на другой. Перед добавлением этих файлов, однако, вы должны выйти из сервера, а затем перезапустить сервер.

Резервное копирование сервера

И наконец, чтобы создать резервную копию всей конфигурации сервера Blocks, просто скопируйте каталог “PIXILAB-Blocks-root” в безопасное место. Чтобы впоследствии восстановить эту резервную копию, закройте сервер и замените каталог “PIXILAB-Blocks-root” на ранее сохраненный.

В. ДОПОЛНИТЕЛЬНЫЕ СКРИПТЫ И ДРАЙВЕРЫ на TypeScript

В дополнение к программированию на основе задач, описанному ранее в этом руководстве Blocks, также поддерживает расширенные сценарии, основанные на формальном языке программирования [TypeScript](#). Этот продвинутый метод может использоваться для обработки очень сложных сценариев, таких как игровые движки, где, как правило, необходимо традиционное программирование.

Подробнее о требованиях и использовании продвинутых скриптов – на [PIXILAB Wiki](#).

Разработка драйверов устройств

Язык программирования TypeScript также используется для разработки пользовательских драйверов устройств, как упоминалось ранее в этом руководстве.

Проведение таких работ требует схожих навыков программирования, см. [PIXILAB Wiki](#).