



Macroscop SDK и API

Опубликовано 16.06.2023

Оглавление

Общие сведения о Macroscop SDK	4
Типовые задачи	5
Краткое описание проектов с примерами использования Macroscop SDK	6
Плагины	7
Регистрация плагинов в Macroscop	7
Сериализуемые классы в плагинах	8
Плагин Действие	8
Плагин Видеоаналитика	11
Плагин Визуализатор	16
Плагин Элемент меню	19
Плагин Процессор событий	21
Плагин Получатель кадров	23
Macroscop API с интерфейсами HTTP и RTSP	37
HTTP-интерфейс для получения данных о системе	37
Получение конфигурации системы	38
Получение списка доступных сеток на клиенте Macroscop	50
Получение списка серверных профилей экрана (видов)	51
Получение информации о текущем профиле экрана в клиенте Macroscop	53
Получение текущего времени сервера Macroscop	55
Получение информации о наличии архива в указанный момент времени	56
Получение списка интервалов с информацией о начале и окончании записи архива	57
Получение информации о состоянии каналов	59
HTTP-интерфейс для получения событий	63
Получение списка всех зарегистрированных в системе событий	64
Получение событий в реальном времени	65
Получение событий из архива	68
Получение списка специальных архивных событий	73
Получение из архива распознанных автомобильных номеров	74
HTTP-интерфейс для выполнения команд сервером Macroscop	76
Включение/выключение записи на канале	76
Изменение времени сервера Macroscop	77
Установка профиля экрана на клиенте	77
Смена сетки на клиенте	78
Установка канала в ячейку сетки	79
Удаление канала из ячейки сетки	80
Очистка сетки камер	80
Постановка канала на охрану	81

Отправка звука на камеру	81
Генерация события из внешней системы	82
HTTP-интерфейс для управления PTZ-возможностями	86
Получение информации о возможностях устройства	87
Получение пресетов устройства	89
Установка пресета	91
«Непрерывное» движение	91
«Непрерывное» изменение фокуса.....	91
«Непрерывный» зум	92
Остановка «непрерывных» команд.....	92
Автофокусировка.....	92
Центрирование.....	93
«Шаговое» движение	93
«Шаговый» зум.....	94
Приближение выделенной области (AreaZoom).....	94
HTTP-интерфейс для получения медиа данных.....	95
Получение кадра	95
Получение «сырого» видео.....	97
Получение перекодированного видео в формате MJPEG	99
Получение фрагмента видео из архива в формате MP4.....	103
HTTP-интерфейс для управления автосменой видов	105
Установка профиля автосмены видов	105
Получение профилей автосмены видов	106
RTSP-интерфейс для получения видео и звука	107
Macroscop API с интерфейсом XML	110
Получение данных счётчика посетителей.....	111
Организация вещания видео на сайт	113
Вещание с помощью HTML5	113
Вещание с помощью Flash (устарело).....	115
Вещание с помощью JavaScript (устарело)	116

Общие сведения о Macroscop SDK

Macroscop SDK – это инструментарий, позволяющий создавать программное обеспечение, именуемое плагинами (внешними модулями), позволяющее расширять существующие функциональные возможности программного комплекса **Macroscop**.

Данный инструментарий предназначен для .NET программистов, желающих создавать плагины для **Macroscop**. Все исходные файлы инструментария и примеров написаны для .NET на языке C#. В качестве среды разработки предполагается использование **Microsoft Visual Studio**. Для понимания данного документа требуется владение терминологией **Macroscop** на уровне опытного пользователя. При необходимости можно обратиться к инструкциям оператора и администратора, поставляемых в комплекте с **Macroscop**.



С версии Macroscop 4.0 используется версия платформы [Net6](#). Плагины для версий 3.6 и ранее должны быть пересобраны под новую версию платформы.

В **Macroscop SDK** каждый плагин представляет собой наследника одного из доступных в инструментарии базовых классов (интерфейсов) и решает определенный ряд задач. На данный момент в инструментарии имеются следующие основные базовые классы (интерфейсы), которые могут быть использованы внешними разработчиками:

Имя плагина	Название	Описание
ExternalAction	Действие	Базовый класс, позволяющий добавлять новые действия для сценариев и планировщика задач по расписанию
VideoAnalyst	Видеоаналитика	Базовый класс для осуществления видеоаналитики на сервере
PluginVisualizer	Визуализатор	Базовый класс визуализатора для графического отображения специфической информации на канале в приложении Macroscop Клиент
IClientPluginMenuItem	Элемент меню	Интерфейс, позволяющий создавать в приложении Macroscop Клиент собственный подпункт в меню Настройка
EventProcessor	Процессор событий	Базовый класс процессора событий. Позволяет регистрировать и генерировать собственные события, получать события из Macroscop , а также выполнять команды в канале. Плагины данного типа могут быть использованы для осуществления интеграции с другими системами.
ICameraServiceProvider	Получатель кадров	Интерфейс получателя кадров с IP-устройств. Позволяет получать с IP-устройств (камер) видео, звук, данные детекции движения; а также управлять поворотными камерами.

Указанные типы базовых классов (интерфейсов), а также некоторые другие вспомогательные сущности подробно рассматриваются в соответствующих главах данного документа. Все плагины существуют и работают в рамках канала **Macroscop**. Таким образом, все экземпляры плагинов по умолчанию изолированы друг от друга, однако имеется возможность обмениваться данными при необходимости через статические поля плагина. Как правило, все плагины, решающие в совокупности одну сложную задачу, находятся в одной сборке .NET. Такая сборка является динамически подключаемой библиотекой (DLL), функционирующей в среде **Macroscop**. Подключение сборок и регистрация плагинов происходит на этапе запуска отдельных компонентов программного комплекса (см. [Регистрация плагинов в Macroscop](#)).

Типовые задачи

1. Интеграция IP-камер

Для подключения IP-устройства (камеры) достаточно реализовать плагин **Получатель кадров**. Сведения о данном типе плагина предоставлены в разделе [Плагин-получатель кадров](#). Шаблон плагина размещен в папке с примерами, в проекте **Camera.csproj**.

2. Интеграция со СКУД, ОПС, POS-терминалами и т.п.

Интеграция может быть выполнена через плагин **Процессор событий**, способный получать события от **Macroscop**, генерировать в **Macroscop** собственные события в процессе взаимодействия с другой системой, выполнять в **Macroscop** команды в канале (включение/выключение записи, установка пресетов, ввод-вывод с камер и др.), получать доступ к архиву **Macroscop**. Сведения о данном типе плагина предоставлены в разделе [Плагин Процессор событий](#). Пример плагина размещен в папке с примерами, в проекте **EP_PluginExample.csproj**. Если требуется получать только видео и звук из **Macroscop**, то можно использовать более простой вариант, рассмотренный в разделе [HTTP-интерфейс для получения видео](#); пример получения видео по HTTP размещен в папке с примерами, в проекте **HttpVideo.csproj**.

3. Видеоаналитика

Любой алгоритм обработки видеопотока может быть реализован с помощью плагина **Видеоаналитика**. Все результаты работы данного плагина представляются событиями, которые могут быть далее интерпретированы плагинами **Визуализатор** и **Элемент меню**. Указанные плагины описаны в разделах [Плагин Видеоаналитика](#), [Плагин Визуализатор](#) и [Плагин Элемент меню](#); пример использования размещен в папке с примерами, в проекте **Analyst_PluginExample.csproj**.

Краткое описание проектов с примерами использования Macroscop SDK

Каждый проект имеет отдельное описание работы в файле README.md.

Также описание и примеры реализации некоторых плагинов находятся в нашем [репозитории на GitHub](#).

Название проекта	Описание
Analyst_PluginExample	Плагин добавляет интеллектуальный модуль Test Event Module , который генерирует событие CounterEvent , отображаемое в журнале событий Macroscop Клиента .
Camera	Плагин добавляет возможность получать тестовое видео с тестовой камеры: Macroscop Конфигуратор -> Камеры -> Добавить камеру -> Производитель = Sample Brand , Тип устройства = Камера, Модель = Sample Camera . В качестве примера, на видео постоянно отображается текст SAMPLE FRAME .
EP_PluginExample	Плагин добавляет модуль External events generating module , который генерирует событие CounterEvent , отображаемое в журнале событий Macroscop Клиента . В событии содержится значение счетчика, которое инкрементируется в соответствии с заданной в настройках частотой.
ExternalAction_PluginExample	Плагин добавляет действие Test external action по событию в сценариях и планировщике задач по расписанию. Действие записывает сообщение в лог Test external action module.log .
HttpInterface	В примере реализовано отображение списка каналов сервера, списка зарегистрированных событий. Также доступно получение событий в реальном времени и специальных архивных событий.
HttpVideo	Пример демонстрирует получение видео (mjpeg) с камеры с помощью HTTP -интерфейса.
MenuItem_PluginExample	Плагин добавляет элемент меню в Клиенте (Левая панель -> Дополнительно -> MENU ITEM EXAMPLE). При нажатии на элемент меню происходит открытие окна, в котором содержатся идентификаторы каналов.
RtVisualizer	Плагин отображает информацию в ячейке наблюдения камеры в Клиенте , при возникновении события тревоги на камере.

Плагины

В данной главе описан процесс создания и регистрации плагинов, а также подробно рассмотрен каждый тип плагина. Наиболее важные фрагменты кода отражены в тексте.

Пример создания простого плагина

Регистрация плагинов в Macroscop

В момент запуска отдельных приложений — компонентов **Macroscop (Сервер/Клиент/Конфигуратор)**; далее в тексте — **хост**), — происходит поиск .NET сборок в папке **Plugins** запускаемого приложения. В каждой найденной сборке должен быть реализован интерфейс **IPlugin**, представленный ниже:

```
public interface IPlugin
{
    /// <summary>
    /// Возвращает уникальный идентификатор модуля
    /// </summary>
    Guid Id { get; }

    /// <summary>
    /// Возвращает название модуля
    /// </summary>
    string Name { get; }

    /// <summary>
    /// Возвращает название производителя модуля
    /// </summary>
    string Manufacturer { get; }

    /// <summary>
    /// Инициализация модуля.
    /// Вызывается хостом на этапе регистрации модуля в системе.
    /// </summary>
    /// <param name="host">Интерфейс хоста</param>
    void Initialize(IPluginHost host);
}
```

Пример реализации данного интерфейса:

```
public class ModuleDef : IPlugin
{
    public Guid Id
    {
        get { return new Guid("17EE3457-8FC2-4C0F-B133-EF11D0C4F38C"); }
    }

    public string Name
    {
        get { return "Модуль поиска оставленных предметов"; }
    }

    public string Manufacturer
    {
        get { return "Macroscop"; }
    }

    public void Initialize(IPluginHost host)
    {
        host.RegisterAnalyst(typeof(AnalystExample));
    }
}
```

```

        host.RegisterExternalEvent(typeof(ObjectLeftEvent));
    }
}

```

Как только указанный интерфейс был обнаружен хостом, вызывается метод инициализации (**Initialize**), в качестве аргумента которому передается интерфейс **IPluginHost**, предоставляющий сервисные методы хоста:

```

public interface IPluginHost : IService
{
    /// <summary>
    /// Получает интерфейс менеджера протоколов
    /// </summary>
    /// <returns></returns>
    IMcLogMgr GetLogManager();

    /// <summary>
    /// Регистрация устройства.
    /// </summary>
    void RegisterDevType(DevType_RegInfo regInfo);

    /// <summary>
    /// Регистрирует внешнее событие
    /// </summary>
    void RegisterExternalEvent(Type eventType);

    /// <summary>
    /// Регистрирует внешнее действие
    /// </summary>
    void RegisterExternalAction(Type actionType);

    /// <summary>
    /// Регистрирует пункт меню в Macroscop клиенте
    /// </summary>
    void RegisterMenuItem(Type menuItemType, List<Guid> requiredPluginIDs);

    // ... Прочие, не показанные здесь, функции регистрации
}

```

Сервисные методы хоста предоставляют следующие возможности:

- Протоколирование работы плагинов с помощью интерфейса **IMcLogMgr**, получаемого соответствующим методом **GetLogManager()**.
- Регистрация в системе плагинов для решения различных задач.

Сериализуемые классы в плагинах

Плагин может содержать сериализуемые классы, например наследники **IAction**, **RawEvent**, классы настроек. Такие классы обычно помечены атрибутом **[Serializable]**. В плагинах такие классы дополнительно необходимо помечать атрибутом **[AlarusSerializable]**.

Плагин Действие

Плагин действие позволяет расширить список функций в сценариях и планировщике задач по расписанию. Для создания такого типа плагина необходимо наследоваться от класса *ExternalAction*:

```

/// <summary>
/// Базовый класс действия.

```



```

/// </summary>
[Serializable][AlarusSerializable]
public abstract class ExternalAction : IAction
{
    [NonSerialized]
    protected IActionHost ActionHost;

    /// <summary>
    /// Инициализация. Вызывается хостом перед началом работы.
    /// </summary>
    /// <param name="host"></param>
    public virtual void Initialize(IActionHost host)
    {
        ActionHost = host;
    }

    /// <summary>
    /// Показывает правильно ли на данный момент
    /// сконфигурировано ли действие
    /// </summary>
    public abstract bool IsConfigurated
    {
        get;
    }

    /// <summary>
    /// Описание на основе текущих настроек
    /// </summary>
    public abstract string Description
    {
        get;
    }

    /// <summary>
    /// Свойство показывает, привязано ли действие к
    /// конкретному каналу. Иными словами, влияет ли действие каким-либо
    /// образом на канал.
    /// </summary>
    public virtual bool IsChannelIndependent
    {
        get
        {
            //по умолчанию действие к каналу никак не привязано
            return true;
        }
    }

    /// <summary>
    /// Запуск действия на выполнение
    /// </summary>
    public abstract void Run(RawEvent rawEvent);

    /// <summary>
    /// Выполнение команд в канале. Заполняется сервером, может использоваться в
    /// методе Run (см. выше)
    /// </summary>
    [NonSerialized]
    public ExecuteCommandDelegate ExecuteCommand;

    /// <summary>
    /// Позволяет генерировать событие, заполняется хостом перед вызовом действия
    /// </summary>

```

```
[NonSerialized]
public GenerateEventDelegate GenerateEvent;
}
```

В классе наследнике требуется задать следующие атрибуты:

- **ActionGUIName** — название действия, которое будет фигурировать в графическом интерфейсе в конфигураторе.
- **GuidAttribute** — идентификатор действия.

Опционально может быть задан атрибут **ActionNeedsEventArgument**, который показывает, что для вызова метода **Run** плагина со стороны сервера обязательно следует передавать объект события. Это также означает, что действие выполняется только по событию и не может выполняться в задачах по расписанию, при выполнении которых события не возникают.

Также требуется определить (переопределить) методы базового класса. Рассмотрим более подробно метод инициализации, в который передается интерфейс **IActionHost** со стороны хоста. Интерфейс выглядит следующим образом:

```
/// <summary>
/// Интерфейс, предоставляющий возможности
/// хоста при инициализации плагина действия.
/// </summary>
public interface IActionHost
{
    /// <summary>
    /// Получение информации о канале,
    /// в котором запущен текущий экземпляр
    /// плагина действия.
    /// </summary>
    /// <returns></returns>
    RawChannelInfo GetChannelInfo();

    /// <summary>
    /// Сохраняет любой сериализуемый объект в конфигурацию
    /// MacroScop. Используется в конфигураторе.
    /// </summary>
    /// <param name="id">Идентификатор объекта</param>
    /// <param name="obj">Объект</param>
    void SaveObject(Guid id, object obj);

    /// <summary>
    /// Получает ранее из сериализованный объект из конфигурации.
    /// Используется в конфигураторе.
    /// </summary>
    /// <param name="id"></param>
    /// <returns></returns>
    object GetObject(Guid id);

    /// <summary>
    /// Отправляет команду плагину, работающего на сервере.
    /// </summary>
    /// <param name="pluginId">Идентификатор плагина</param>
    /// <param name="channelsIds">Идентификаторы каналов</param>
    /// <param name="command">Команда</param>
    /// <returns>Возвращает результат от каждого канала. Ключ - идентификатор канала.
    /// Значение - результат выполнения команды.</returns>
    Dictionary<Guid, object> SendCommandToPlugin(Guid pluginId, List<Guid>
channelsIds, object command);
}
```

Данный интерфейс позволяет получить информацию о канале, к которому привязан экземпляр плагина. Эта информация включает в себя имя канала и его идентификатор. Идентификатор должен быть использован при выполнении команд в канале (см. делегат **ExecuteCommand**). Подробная информация о доступных командах приведена в разделе [Плагин Процессор событий](#). Кроме того, данный интерфейс с помощью методов **SaveObject** и **GetObject** позволяет сохранять и загружать любой сериализуемый объект по идентификатору на этапе задания пользователем конфигурации **Macroscop**. Такой механизм позволяет сохранять и извлекать настройки, которые являются одинаковыми для всех экземпляров плагинов. Свойство **IsConfigured** показывает, правильно ли пользователь сконфигурировал действие в графическом интерфейсе; если неправильно, то конфигурацию невозможно будет применить до тех пор, пока ошибки не будут исправлены.

Для регистрации **Действие** в системе, необходимо на этапе загрузки сборки с плагином в методе инициализации класса, реализующего интерфейс **IPlugin**, вызвать метод **RegisterExternalAction** интерфейса хоста (см. [Регистрация плагинов в Macroscop](#)).

Для действий, имеющих настройки в графическом интерфейсе, необходимо добавить атрибут **ActionAssemblyPluginHasSettingsControl** в файл **AssemblyInfo.cs**.

Плагин Видеоаналитика

Данный тип плагина осуществляет покадровый анализ видеопотока, что позволяет создавать сервисные детекторы, например, детектор оставленных предметов, детектор саботажа и др. Для создания такого типа плагина, необходимо создать наследника базового класса **VideoAnalyst**:

```

/// <summary>
/// Класс видеоаналитика. Используется для обработки кадров и карт движения.
/// </summary>
public abstract class VideoAnalyst : IDisposable
{
    /// <summary>
    /// Инициализация аналитика. Вызывается хостом перед началом процесса обработки.
    /// </summary>
    /// <param name="id">Идентификатор канала</param>
    /// <param name="analystToolSet">Тул сет аналитики</param>
    /// <param name="pluginEnv">Настройки плагина</param>
    public abstract void Initialize(Guid id, IPluginAnalystToolSet analystToolSet,
    PluginEnvironment pluginEnv);

    /// <summary>
    /// Метод обработки кадра и карты движения. Вызывается хостом.
    /// </summary>
    /// <param name="image">Кадр</param>
    /// <param name="motionMap">Карта движения, может быть null</param>
    /// <param name="background">Фон от детектора движения. Равен null, если
    /// NeedBackground == false</param>
    public abstract void Process(ImageData image, MotionMap motionMap,
    BackgroundImage background);

    /// <summary>
    /// Генерирует в канале ранее зарегистрированное внешнее событие. Заполняется
    /// хостом. Вызывается аналитиком.
    /// </summary>
    public GenerateEventDelegate GenerateEvent;

    /// <summary>
    /// Поддерживает ли аналитик работу с частично декодированными кадрами.
    /// True означает, что на вход могут подаваться уменьшенные видеокдры,
    /// False означает, что на вход всегда будут подаваться видеокдры с оригинальным
    /// разрешением.

```

```

/// </summary>
public virtual bool SupportsPartlyDecodedFrames => false;

public virtual bool NeedBackground => false;

public virtual TimeSpan MinTimeBetweenFrames => TimeSpan.Zero;

public virtual TimeSpan MaxTimeBetweenFrames => TimeSpan.MaxValue;

protected ServerRole ServerRoleForChannel { get; set; }

/// <summary>
/// Поддерживаемый формат пикселя
/// </summary>
public virtual VAPixelFormat PixelFormat => VAPixelFormat.BGR24;

/// <summary>
/// Выполняет команду.
/// </summary>
/// <param name="cmdObj"></param>
/// <returns></returns>
public virtual object ProcessCommand(object cmdObj)
{
    return null;
}

public SendServerCommandDelegate SendServerCommand;

/// <summary>
/// Освобождение ресурсов
/// </summary>
public abstract void Dispose();

/// <summary>
/// Изменяет общие или специфические настройки аналитика, если это необходимо.
/// Вызов метода должен приводить к появлению пользовательского окна настройки.
/// Вызывается хостом (конфигуратором).
/// </summary>
/// <param name="channel"></param>
/// <param name="settingsHost"></param>
/// <param name="settings">Текущие параметры аналитика</param>
/// <returns>Новые параметры аналитика</returns>
public abstract PluginSettings SetSettings(Guid channel, ISettingsHost
settingsHost, PluginSettings settings);

public virtual object GetAdditionalInfo()
{
    return null;
}

public virtual object GetAdditionalInfo(params object[] parameters)
{
    return null;
}

/// <summary>
/// Возвращает конфигурацию в виде массива байт
/// </summary>
/// <returns></returns>
public virtual byte[] GetBinarySettings()
{
    return null;
}

```

```

}

/// <summary>
/// Изменяет общие или специфические настройки аналитика, если это необходимо.
/// </summary>
/// <param name="binarySettings">настройки аналитика в виде массива байт</param>
public virtual void SetSettings(byte[] binarySettings)
{
}

public virtual List<VideoAnalystZone> GetZones(object channelSpecificSettings)
{
    return new List<VideoAnalystZone>();
}

public virtual bool NeedReferencedImage()
{
    return false;
}

/// <summary>
/// Нужны ли вообще аналитику видеок cadры? (Некоторые аналитики, например,
/// 3D-подсчет сейчас получают с камеры готовые данные. В этом случае видео
тянуть
/// не надо).
/// </summary>
/// <returns></returns>
public virtual bool NeedVideoFrames()
{
    return true;
}

/// <summary>
/// Конвертирует настройки аналитика из Json <paramref name="settingsJson"/> с
/// учетом текущих настроек из
/// конфигурации <paramref name="currentSettings"/> и возвращает новые настройки
/// аналитика.
/// </summary>
/// <param name="settingsJson">Новые настройки аналитика в формате Json</param>
/// <param name="currentSettings">Текущие настройки аналитика</param>
/// <param name="converter">Json конвертер</param>
/// <returns>Новые настройки аналитика</returns>
public virtual PluginSettings ConvertJsonSettingsToPluginSettings(string
settingsJson,
    PluginSettings currentSettings, IJsonConverter converter) =>
default(PluginSettings);

/// <summary>
/// Конвертирует текущие настройки аналитика <paramref name="currentSettings"/> в
/// Json и возвращает их.
/// </summary>
/// <param name="currentSettings">Текущие настройки аналитика</param>
/// <param name="converter">Json конвертер</param>
/// <returns>Настройки в формате Json</returns>
public virtual string ConvertToJsonSettings(PluginSettings currentSettings,
IJsonConverter converter) => null;
}

```

Класс наследник должен переопределить все абстрактные методы базового класса и, при необходимости, виртуальные свойства. Кроме того, производный класс должен иметь обязательный атрибут **PluginGUINameAttribute**, содержащего название аналитики, отображаемой в графической оболочке приложения **Macroscop Конфигуратор**. Если аналитика может быть настроена в конфигураторе, необходимо реализовать метод настройки **SetSettings**, и указать атрибут **PluginHasSettingsAttribute**.

Данные изображений подаются на вход аналитики в виде класса **ImageData**.

Поскольку каждый аналитический плагин прикрепляется пользователем к тому или иному каналу, в конфигураторе **Macroscop** объект настроек **PluginSettings** состоит из 2 частей:

- 1) настройки, связанные с текущим каналом безопасности
- 2) общие настройки аналитика, не зависящие от выбранного канала безопасности.

```
public struct PluginSettings
{
    /// <summary>
    /// Специфические настройки плагина, связанные с текущим каналом. Может быть null.
    /// Объект настроек должен быть сериализуемым.
    /// </summary>
    public object channelSpecificSettings;

    /// <summary>
    /// Общие настройки плагина. Не зависят от текущего канала. Может быть null.
    /// Объект настроек должен быть сериализуемым.
    /// </summary>
    public object generalSettings;
}
```

Если настройки аналитики едины для всех каналов, то достаточно заполнять только объект общих настроек. В противоположном случае, когда все настройки аналитики привязываются к конкретному каналу, достаточно заполнять только объект специфических настроек канала. Объекты общих и специфических настроек являются пользовательскими. Единственным требованием для них является возможность их сериализации, поскольку все настройки аналитики хранятся в общей конфигурации **Macroscop**.

Рассмотрим также метод **ProcessCommand**, который позволяет аналитике выполнять команды от плагина **Элемент меню** (см. [Плагин Элемент меню](#)). Данный метод получает команду в виде сериализуемого объекта, который формируется на стороне плагина **Элемент меню**. В качестве результата данный метод также должен возвращать сериализуемый объект, который впоследствии получит и обработает плагин **Элемент меню**. Такой механизм позволяет реализовать клиент-серверное взаимодействие, поскольку плагин **Видеоаналитика** работает на сервере, а плагин **Элемент меню** — всегда в клиенте.

В процессе обработки видеок кадров аналитика должна передавать результаты своего анализа серверу с помощью генерации событий. Для этого необходимо заранее зарегистрировать на стороне хоста (см. [Регистрация плагинов в Macroscop](#)) внешнее пользовательское событие, содержащее описание всех необходимых полей для интерпретации результатов работы аналитика. Регистрируется событие с помощью метода **RegisterExternalEvent** интерфейса **IPluginHost** на этапе инициализации модуля. Пользовательское событие должно быть унаследовано от базового класса **RawEvent**:

```
/// <summary>
/// Родительский класс в иерархии событий, происходящих в канале.
/// </summary>
[Serializable][AlarusSerializable]
public abstract class RawEvent : IDescribable
{
```

```

/// <summary>
/// Временная метка события.
/// </summary>
[OptionalField]
public DateTime EventTime;

/// <summary>
/// Комментарий к событию.
/// </summary>
[OptionalField]
public string Comment;

public virtual EventTextDescription GetDescription(IToStringConverter converter)
{
    return new EventTextDescription("");
}

public virtual object GetAttachments()
{
    return null;
}

/// <summary>
/// Создание сырого события
/// </summary>
protected RawEvent()
{
    EventTime = DateTime.UtcNow;
}
}

```

Каждое пользовательское событие должно иметь ряд обязательных атрибутов:

- **GuidAttribute** — определяет в явном виде уникальный идентификатор события;
- **Serializable** и **AlarusSerializable** — позволяют выполнять сериализацию события.

Кроме того, имеется ряд необязательных (опциональных) атрибутов:

- **EventLocalizedName** — задает название события в графическом интерфейсе хоста. Если атрибут не задан, то событие не будет отображаться в конфигураторе в разделе сценариев;
- **EventSaveable** — задает название таблицы в базе данных, в которую будут сохраняться экземпляры события; атрибут должен использоваться в случае, если у события есть поля, которые должны быть сохранены в БД (важно отметить, что свойство события **SaveMode** должно принимать в этом случае значения **Special**, **UnifiedLog** или **SpecialAndUnifiedLog**). Свойство события **GenerationFrequency** задает частоту генерации данного события, требует параметр **EventGenerationFrequency** (см. ниже). Данное свойство влияет на работу сервера при сохранении событий в БД и имеет рекомендательный характер. Если свойство не было указано, то по умолчанию используется режим **Middle**. Этот режим является предпочтительным. Режим **Low** не рекомендуется для использования, поскольку данные события записываются в особую, критичную для функционирования, базу данных;
- **EventGeneratesAlarmByDefault** — событие по умолчанию является тревожным, что означает автоматическую привязку к событию действия **Генерация тревоги** при создании нового канала в конфигураторе **Macroscop**.

Если пользовательское событие имеет поля, которые должны сохраняться в БД, необходимо для каждого из полей указывать атрибут **EventFieldSaveableOrScenariosUsable**. В качестве параметров атрибут требует указать **orderNum** (порядковый номер поля, отсчет начинается с 0) и **indexable** (флаг, указывающий, нужно ли индексировать данное поле).

Поддерживаются следующие типы полей события, к которым применимы указанные выше атрибуты: **int, bool, long, double, DateTime, string, Guid, byte[]**.

Свойство **SaveMode** определяет, будет ли событие сохраняться в БД. Событие может храниться как в базовой таблице, в которой находятся только поля класса **RawEvent**, так и в специальной, содержащей все поля события. Также имеется возможность сохранять событие в обе таблицы. Использование базовой таблицы позволяет фиксировать сам факт возникновения события в системе. В будущем пользователь может читать из базовой таблицы события штатными средствами **Macroscop**.

Пользовательское событие может быть представлено следующим образом:

```
[GuidAttribute("389EDCE2-54BB-4C2C-9984-51B7516A5DDF")]
[EventLocalizedName("Оставлен предмет")]
[EventSaveable(EventSaveMode.SpecialAndUnifiedLog, "objectleft")]
[EventGeneratesAlarmByDefault]
[Serializable][AlarusSerializable]
public class ObjectLeftEvent : RawChannelEvent
{
    [EventFieldSaveableOrScenariosUsable(0, true)]
    [EventFieldLocalizedName("Предмет")]
    private string objectName;

    public ObjectLeftEvent(string objectName)
    {
        this.objectName = objectName;
    }

    public override EventTextDescription GetDescription(IToStringConverter converter)
    {
        var eventText = "Оставлен предмет";
        var eventDescription = new EventTextDescription(eventText);

        return eventDescription;
    }
}
```

Для регистрации плагина **Видеоаналитика** необходимо на этапе загрузки сборки с плагином в методе инициализации класса, реализующего интерфейс **IPlugin**, вызвать метод **RegisterAnalyst** интерфейса хоста (см. [Регистрация плагинов в Macroscop](#)).

Плагин Визуализатор

Плагин данного типа позволяет графически отображать информацию (например, рамки движущихся объектов, распознанные номера, лица и др.) содержащуюся в событиях, которые поступают в каналы приложения **Macroscop Клиент**. Для создания данного типа плагина необходимо создать наследника класса **PluginVisualiser**:

```
/// <summary>
/// Класс визуализатора.
/// </summary>
public abstract class PluginVisualiser
{
    public abstract DrawingVisual NullableDrawingVisual { get; }

    /// <summary>
```



```

/// Панель для отрисовки примитивов, текста и другой информации на канале.
/// </summary>
protected IDrawingPanel DrawPanel;

/// <summary>
/// Контейнер графических элементов. Позволяет размещать отдельные
/// UserControl'ы в канале.
/// </summary>
protected Canvas ControlsContrainer;

protected IPluginClientToolSet PluginClientToolset;

public readonly Guid AnalystPluginId;

public bool ArchiveMode;

public virtual IVideoTransformer VideoTransformer
{
    get { return null; }
}

protected PluginVisualiser(Guid analystPluginId)
{
    AnalystPluginId = analystPluginId;
    ArchiveMode = false;
}

/// <summary>
/// заданы ли пользователем настройки визуализатора
/// </summary>
protected bool _hasUserSettings;

public bool HasUserSettings
{
    get { return _hasUserSettings; }
}

/// <summary>
/// Инициализация визуализатора. Вызывается хостом.
/// </summary>
public virtual void Initialize(Guid channelId, IPluginClientToolSet
pluginClientToolset, IPluginVisualizerSet visualiserSet)
{
    PluginClientToolset = pluginClientToolset;
    DrawPanel = visualiserSet.DrawingPanel;
    ControlsContrainer = visualiserSet.ControlsContrainer;
    RegisterPluginMenuAction = visualiserSet.AddMenuActionHandler;

    if (visualiserSet.VisualizeParams.ParamsRef == null)
        _hasUserSettings = false;
    else
        _hasUserSettings = true;
}

/// <summary>
/// Обработка события визуалайзером. Специфическая отрисовка результатов события.
/// </summary>
/// <param name="channelId">Идентификатор канала</param>
/// <param name="chEv">Событие.</param>
/// <param name="isAlarm">Является ли событие тревожным</param>
public abstract void ProcessEvent(Guid channelId, RawEvent chEv, bool isAlarm);

```

```

/// <summary>
/// Делегат для регистрации подпункта во всплывающем меню канала в клиенте
/// Macroscop.
/// Заполняется хостом. Вызывается визуализатором.
/// </summary>
protected RegisterPluginMenuActionHandler RegisterPluginMenuAction;

/// <summary>
/// Очистка всего, что нарисовано визуализатором.
/// </summary>
public abstract void Clear();
/// <summary>
/// Освобождение всех ресурсов. В перегружаемых методах в наследниках обязательно
/// вызывать Release базового класса.
/// </summary>
public virtual void Release()
{
    // сделал, это чтобы очищались ресурсы визуализаторов, т.к. по факту
реализации
    // не перегружают Release, а копипастить неохота.
    Clear();

    DrawPanel = null;
    ControlsContainer = null;
    PluginClientToolset = null;
}

public void SetPresentationModeIfNeeded()
{
    if (!HasUserSettings)
        SetPresentationMode();
}

/// <summary>
/// Переопределить в визуализаторе плагина, если требуется автоматическое
/// отображение результатов/настроек плагина в демо-версии
/// </summary>
protected virtual void SetPresentationMode()
{
}

public virtual void SetDigitalZoomStatus(bool isDigitalZoomActive)
{
}
}

```

Каждый плагин **Визуализатор** должен определять метод **ProcessEvent**, на вход которому передается очередное событие, поступившее в канал. Все события генерируются **Macroscop** или другими плагинами.

Плагин может визуализировать любые типы событий, а их фильтрацию можно осуществить с помощью оператора **if** и ключевого слова **is**, например:

```

if (chEv is CounterEvent)
{
    ...
}

```

Визуализатор может регистрировать свой подпункт во всплывающем меню, отображаемом при щелчке правой кнопки мыши на канале в приложении **Macroscop Клиент**. Это позволяет изменять логику работы визуализатора в зависимости от предпочтений пользователя. Данная возможность предоставляется делегатом **RegisterPluginMenuItemAction**.

Для регистрации плагина-визуализатора в системе, необходимо на этапе загрузки сборки с плагином в методе инициализации класса, реализующего интерфейс **IPlugin**, вызвать метод **RegisterRTVisualizer** интерфейса хоста (см. [Регистрация плагинов в Macroscop](#)).

Плагин Элемент меню

Плагин данного типа позволяет создавать собственный графический интерфейс, открываемый через подпункт меню **Дополнительно** приложения **Macroscop Клиент**.

Типичное применение данного плагина заключается в организации клиент-серверного взаимодействия с другими плагинами. Например, плагин может обрабатывать результаты работы аналитических плагинов, работающих на сервере. Для создания плагина-элемента меню необходимо реализовать интерфейс **IClientPluginMenuItem**:

```
public interface IClientPluginMenuItem
{
    void ShowWindow(IPluginClientToolSet pluginToolSet);

    string Header { get; }

    string ImageName { get; }

    bool HasActivePlugins(IPluginClientToolSet pluginToolSet);
}
```

В классе-наследнике необходимо определить метод **ShowWindow**, в который со стороны хоста (приложения **Macroscop Клиент**) передается интерфейс с сервисными функциями. Эти функции позволяют получить идентификаторы каналов и их имена в текущей конфигурации. Кроме того, они предоставляют доступ к архиву **Macroscop**, возможность отправлять команды аналитическим плагинам на сервере и получать результат

их исполнения. Имеется возможность подписываться на любые события, возникающие в системе. Интерфейс представляется следующим образом:

```
/// <summary>
/// Интерфейс предоставляемый хостом для
/// доступа в архив, для отправки команд, для
/// подписки на события в системе.
/// </summary>
public interface IPluginClientToolSet
{
    /// <summary>
    /// Получает информацию о том, доступно ли текущему пользователю редактирование
    /// данных в клиенте в интеллектуальных плагинах
    /// </summary>
    bool CanAccessEditingAnalystPluginsInClient { get; }

    /// <summary>
    /// Получает информацию о том, доступно ли текущему пользователю доступ к базам
    /// данных
    /// </summary>
    bool CanUseDatabase { get; }

    /// <summary>
```

```

/// Получает информацию о том, доступны ли текущему пользователю отчеты
/// </summary>
bool CanUseReports { get; }

/// <summary>
/// Получает интерфейс для чтения конфигурации
/// </summary>
/// <returns></returns>
IPluginsConfigReader ConfigReader { get; }

/// <summary>
/// Получает интерфейс для работы с архивом
/// </summary>
/// <returns></returns>
IArchiveEventsReader ArchiveEventsReader { get; }

/// <summary>
/// Предоставляет перевод идентификаторов сущностей(каналы, пользователи, модули
/// аналитики, события)
/// в строковое представления для GUI.
/// </summary>
IToStringConverter ToStringConverter { get; }

IImageResizer ImageResizer { get; }

/// <summary>
/// Отправляет команду плагину,
/// работающего на сервере.
/// </summary>
/// <param name="pluginId">Идентификатор плагина</param>
/// <param name="channelsId">Идентификаторы каналов</param>
/// <param name="cmdObj">Команда</param>
/// <returns>Возвращает результат от каждого канала. Ключ - идентификатор канала.
/// Значение - результат выполнения команды.</returns>
Dictionary<Guid, object> SendChannelsCommand(Guid pluginId, List<Guid>
channelsId, object cmdObj);

/// <summary>
/// Устанавливает/удаляет обработчик событий, возникающих в канале.
/// </summary>
/// <param name="subscrId">Идентификатор подписчика</param>
/// <param name="channelId">Идентификатор канала</param>
/// <param name="eventsHandler">Обработчик. Если null, то ранее установленный
/// обработчик удаляется.</param>
void SetEventsHandler(Guid subscrId, Guid channelId, EventHandler
eventsHandler);

/// <summary>
/// Контроллер, отвечающий за взаимодействие различных GUI подсистем внутри
/// плагина.
/// </summary>
IGuiScreensInteractionController InteractionController { get; }
}

```

Свойство **ArchiveEventsReader** предоставляет интерфейс доступа к архиву. Подробные сведения о данном интерфейсе размещены в исходных кодах **Macroscop SDK**.

Метод **SendChannelsCommand** отправляет команду разным экземплярам плагина одного и того же типа по указанному списку каналов и получает результаты исполнения команды. Метод **SetEventsHandler** устанавливает или удаляет обработчик событий на заданном канале.

В методе **ShowWindow** должна быть реализована логика работы с пользователем через графический интерфейс. Данный метод вызывается в момент щелчка клавиши мыши (клавиатуры) по пункту меню, связанному с плагином.

Для регистрации плагина **Элемент меню** необходимо на этапе загрузки сборки с плагином в методе инициализации класса, реализующего интерфейс **IPlugin**, вызвать метод **RegisterMenuItem** интерфейса хоста (см. [Регистрация плагинов в Macroscop](#)).

Плагин Процессор событий

Плагин **Процессор событий** позволяет получать и обрабатывать сигналы от сторонних систем. В процессе обработки сигналов данный тип плагина может как выполнять команды в канале, в котором он существует, так и генерировать в этом канале события. Данный плагин создается путем наследования от базового класса **EventProcessor**:

```

/// <summary>
/// Класс плагина для обработки событий системы, генерации команд и своих событий
/// </summary>
public abstract class EventProcessor : IDisposable
{
    /// <summary>
    /// Инициализация. Вызывается хостом.
    /// </summary>
    /// <param name="settings">Настройки плагина</param>
    public abstract void Initialize(PluginSettings settings);

    /// <summary>
    /// Генерирует в канале ранее зарегистрированное внешнее событие. Заполняется
    /// хостом. Вызывается плагином.
    /// </summary>
    public GenerateEventDelegate GenerateEvent;

    /// <summary>
    /// Запускает заданную команду на выполнение в канале. Заполняется хостом.
    /// Вызывается плагином.
    /// </summary>
    public ExecuteCommandExDelegate ExecuteCommandSync;

    /// <summary>
    /// Позволяет подписываться на события, происходящие в канале. Заполняется
    плагинном
    /// при необходимости на этапе инициализации.
    /// </summary>
    public ReceiveEventDelegate OnChannelEventReceived;

    /// <summary>
    /// Изменяет общие или специфические настройки, если это необходимо.
    /// Вызов метода должен приводить к появлению пользовательского окна настройки.
    /// Вызывается хостом (конфигуратором).
    /// </summary>
    /// <param name="settings">Текущие настройки плагина.</param>
    /// <returns>Новые настройки плагина.</returns>
    public abstract PluginSettings SetSettings(PluginSettings settings);

    public abstract void Dispose();
}

```

Производный класс должен иметь обязательный атрибут **PluginGUINameAttribute**, содержащий название плагина, отображаемое в приложении **Macroscop Конфигуратор**. Если плагин может быть настроена в конфигураторе, следует реализовать метод настройки **SetSettings** и указать атрибут **PluginHasSettingsAttribute**.

В метод инициализации **Initialize** со стороны хоста передается объект настроек плагина. Поскольку каждый плагин прикрепляется пользователем к тому или иному каналу, объект настроек **PluginSettings** состоит из 2 частей:

- 1) настройки, связанные с текущим каналом безопасности;
- 2) общие настройки, не зависящие от выбранного канала безопасности.

```
public struct PluginSettings
{
    /// <summary>
    /// Специфические настройки плагина, связанные с текущим каналом.
    /// Объект настроек должен быть сериализуемым.
    /// </summary>
    public object channelSpecificSettings;

    /// <summary>
    /// Общие настройки плагина. Не зависят от текущего канала. Может быть null.
    /// Объект настроек должен быть сериализуемым.
    /// </summary>
    public object generalSettings;
}
```

Если настройки едины для всех каналов, то заполняется объект общих настроек. В случае, когда все настройки плагина привязываются к конкретному каналу, достаточно заполнять только объект специфических настроек канала. Объекты общих и специфических настроек являются пользовательскими (объект настройки канала всегда отличен от null); одним из требований к ним является возможность сериализации, поскольку все настройки плагина хранятся в общей конфигурации **Macroscop**. Если плагину в качестве результата своей работы необходимо выполнить определенную команду в канале (например, включить/выключить запись, установить пресет на камере, повернуть камеру и т.д.), требуется создать объект команды. На текущий момент реализованы следующие команды:

- **RawEnableRecordingCommand** — включает запись в канале, опционально указывается интервал записи;
- **RawDisableRecordingCommand** — выключает запись в канале;
- **RawGoToPresetPtzCommand** — устанавливает пресет на камере;
- **RawGoHomePtzCommand** — устанавливает домашнее положение камеры;
- **RawStopPtzCommand** — останавливает выполнение ptz команд на камере;
- **RawMovePtzCommand** — перемещение на шаг;
- **RawZoomPtzCommand** — относительное приближение (зум);
- **RawStartMovePtzCommand** — непрерывное (желательно плавное) движение;
- **RawMoveToPtzCommand** — поворачивает камеру таким образом, что указанная точка оказывается в центре области кадра;
- **RawSetDigitalOutputCommand** — устанавливает уровень сигнала на выходе камеры;
- **RawSetDigitalPulsesCommand** — генерирует последовательность импульсов на выходе камеры;

Плагин **Процессор событий** может подписываться на события, возникающие в канале. Для этого на этапе своей инициализации плагин должен заполнить делегат **OnChannelEventReceived**. Кроме того, плагин может генерировать собственные события в канале.

Для регистрации плагина **Процессор событий** необходимо на этапе загрузки сборки с плагином в методе инициализации класса, реализующего интерфейс **IPlugin**, вызвать метод **RegisterEventProcessor** интерфейса хоста (см. [Регистрация плагинов в Macroscop](#)).

Плагин Получатель кадров

Плагин данного типа позволяет получать видео и звук с IP-устройств (камер), данные о детекции движения, управлять поворотными камерами, управлять входами/выходами (I/O) IP-устройств. Для решения подзадачи получения кадров необходимо реализовать интерфейс **ICameraServiceProvider**:

```

/// <summary>
/// Интерфейс получения кадров реального времени.
/// Используется для создания плагинов, получающих кадры
/// с IP-камер.
/// </summary>
public interface ICameraServiceProvider
{
    /// <summary>
    /// Асинхронное получение информации об устройстве
    /// </summary>
    /// <param name="callback"></param>
    /// <param name="state"></param>
    /// <returns></returns>
    IAsyncResult BeginGetCapabilities(AsyncCallback callback, object state);

    /// <summary>
    /// Завершает асинхронный запрос на получение информации об устройстве
    /// </summary>
    /// <param name="asyncResult"></param>
    /// <returns></returns>
    DeviceCapabilities EndGetCapabilities(IAsyncResult asyncResult);

    /// <summary>
    /// Обработчик события о приходе нового кадра.
    /// Вызывается реализующей интерфейс стороной.
    /// На событие подписывается хост.
    /// </summary>
    event NewRawFrameEventHandler NewRawFrame;

    /// <summary>
    /// Обработчик событий. Вызывается реализующей интерфейс стороной.
    /// На обработчик подписывается хост.
    /// </summary>
    event NewRawEventHandler NewEvent;

    /// <summary>
    /// Является ли поток активным
    /// </summary>
    /// <param name="streamType">Тип потока</param>
    /// <returns></returns>
    bool IsStreamActive(ChannelStreamTypes streamType);

    /// <summary>
    /// Запускает поток указанного типа для получения кадров
    /// </summary>
    /// <param name="channelStreamType"></param>
    void StartStream(ChannelStreamTypes channelStreamType);

    /// <summary>
    /// Останавливает поток указанного типа

```

```

/// </summary>
/// <param name="channelStreamType"></param>
void StopStream(ChannelStreamTypes channelStreamType);

/// <summary>
/// Отправляет звук на устройство (случай реализации дуплексного звука).
/// </summary>
/// <param name="frame"></param>
void SendSound(RawSoundFrame frame);

/// <summary>
/// Интерфейс работы с PTZ
/// </summary>
/// <returns></returns>
IPtzController GetPtzController();

/// <summary>
/// Интерфейс работы с цифровыми выходами
/// </summary>
/// <returns></returns>
IDigitalOutputsController GetDigitalOutputsController();

/// <summary>
/// Интрейфес для работы с архивом устройства
/// </summary>
/// <returns></returns>
IDeviceArchiveController GetDeviceArchiveController();

/// <summary>
/// Освобождает все ресурсы. Закрывает все потоки.
/// </summary>
void Release();
}

```

При реализации данного интерфейса необходимо создать механизм работы с потоками данных. Потоки данных представляют собой или последовательность кадров определенного типа, или последовательность событий. В текущей версии **Macroscop SDK** имеются следующие типы потоков данных:

```

/// <summary>
/// Типы потоков канала.
/// </summary>
[Flags]
public enum ChannelStreamTypes
{
    /// <summary>
    /// Главный поток видео
    /// </summary>
    [Description("MacroscopSDK.IpDevices.ChannelStreamTypes.MainVideo")]
    MainVideo = 1,

    /// <summary>
    /// Альтернативный поток видео
    /// </summary>
    [Description("MacroscopSDK.IpDevices.ChannelStreamTypes.AlternativeVideo")]
    AlternativeVideo = 2,

    /// <summary>
    /// Поток звука, идущий с камеры.
    /// </summary>
}

```



```

[Description("MacroscopSDK.IpDevices.ChannelStreamTypes.MainSound")]
MainSound = 4,

/// <summary>
/// Альтернативный звуковой поток
/// </summary>
[Description("MacroscopSDK.IpDevices.ChannelStreamTypes.AlternativeSound")]
AlternativeSound = 8,

/// <summary>
/// Обратный поток звука.
/// </summary>
[Description("MacroscopSDK.IpDevices.ChannelStreamTypes.OutputSound")]
OutputSound = 16,

/// <summary>
/// Поток данных детекции движения.
/// </summary>
[Description("MacroscopSDK.IpDevices.ChannelStreamTypes.MotionDetection")]
MotionDetection = 32,

/// <summary>
/// Поток данных от системы ввода-вывода камеры
/// </summary>
[Description("MacroscopSDK.IpDevices.ChannelStreamTypes.IO")]
IO = 64,

/// <summary>
/// Архивное видео
/// </summary>
[Description("MacroscopSDK.IpDevices.ChannelStreamTypes.ArchiveVideo")]
ArchiveVideo = 128,

/// <summary>
/// Архивный звук
/// </summary>
[Description("MacroscopSDK.IpDevices.ChannelStreamTypes.ArchiveSound")]
ArchiveSound = 256,
}

```

В зависимости от возможностей подключаемого устройства, а также от настроек канала в конфигураторе, необходимо генерировать те или иные потоки данных.

Потоки данных **MainVideo** и **AlternativeVideo** состоят из видеок кадров **RawVideoFrame**, получаемых с камеры.

Потоки данных **MainSound** и **AlternativeSound**, **OutputSound** состоят из последовательности звуковых кадров **RawSoundFrame**.

Запуск потоков данных происходит со стороны хоста с помощью метода **StartStream**, в котором производится необходимая инициализация очередного потока. Метод должен сразу же возвращать управление хосту, а длительные операции (операции ввода/вывода) выполнять в отдельных потоках. Остановка потоков и проверка их активности методами **StopStream** и **IsStreamActive** также вызывается со стороны хоста и должны выполняться немедленно без выполнения долгосрочных операций. Результаты своей работы потоки должны возвращать хосту через вызовы обработчиков кадров (**NewRawFrame**) и событий (**NewEvent**).

Например, если IP-устройство шлет MJPEG-кадры, то поток данных **MainVideo** (или **AlternativeVideo**) должен вызывать **NewRawFrame** и в качестве одного из аргументов передавать видеок кадр **RawMJPEGFrame**:

```

/// <summary>

```

```

/// MJPEG кадр
/// </summary>
[Serializable][AlarusSerializable]
public class RawMJPEGFrame : RawVideoFrame
{
    private bool _isNMjpegFrame;

    public bool IsNMjpegFrame
    {
        get { return _isNMjpegFrame; }
        set { _isNMjpegFrame = value; }
    }

    public RawMJPEGFrame()
    {
    }

    public RawMJPEGFrame(IReferencedBytes refBytes)
    {
        UndecodedRefData = refBytes;
    }
}

```

Аналогично для потока данных **MainSound** (или **AlternativeSound**) и звука в формате G.711U, необходимо передавать кадр **RawG711UFrame**:

```

/// <summary>
/// Кадр стандарта G.711U
/// </summary>
[Serializable][AlarusSerializable]
public class RawG711UFrame : RawSoundFrame
{
    public RawG711UFrame()
    {
    }

    /// <summary>
    /// Данные кадра
    /// </summary>
    public RawG711UFrame(IReferencedBytes refBytes)
    {
        UndecodedRefData = refBytes;
        SamplesPerSecond = 8000;
        BitsPerSample = 16;
        Channels = 1;
        Bitrate = 64000;
    }

    /// <summary>
    /// Размер фрейма должен быть выравнен на размер минимального блока данных,
    /// с которыми работает декодер
    /// </summary>
    /// <returns>Возвращает гранулярность фрейма</returns>
    public static int Granularity()
    {
        return 80;
    }
}

```

Пример создания MJPEG-кадра доступен в проекте **Camera.csproj**.

В случае если в процессе получения потоков данных произошел обрыв связи с IP- устройством, плагин **Получатель кадров** должен уведомить об этом хост путем генерации события **NoDataConnectionDeviceEvent** с обязательно заполненным полем **StreamTypesMask**, показывающим, в каких потоках данных произошел обрыв соединения.

Для регистрации получателя кадров в системе необходимо заполнить регистрационную информацию устройства **DevType_RegInfo** и регистрационную информацию потока получения данных **MediaStream_RegInfo**. Ниже приведено описание класса **DevType_RegInfo**:

```

/// <summary>
/// Регистрационная информация устройства.
/// используется при регистрации плагина,
/// получающего кадры.
/// </summary>
public class DevType_RegInfo
{
    /// <summary>
    /// Идентификатор устройства.
    /// </summary>
    public Guid DeviceTypeGuid;

    /// <summary>
    /// Альтернативные идентификаторы устройства, если есть
    /// </summary>
    public Guid[] DeviceAlternativeGuids;

    /// <summary>
    /// Имя производителя.
    /// </summary>
    public string DevTypeBrandName;

    /// <summary>
    /// Имя устройства.
    /// </summary>
    public string DevTypeModelName;

    /// <summary>
    /// Список возможностей устройства в целом.
    /// </summary>
    public DevType_Capabilities Capabilities;

    /// <summary>
    /// Панорамные режимы.
    /// </summary>
    public PanoramicMode[] PanoramicModes;

    /// <summary>
    /// Список доступных разрешений для данного устройства.
    /// </summary>
    public List<Resolution> AvailableResolutions = new List<Resolution>();

    /// <summary>
    /// Делегат, позволяющий хосту изменять настройки на камере/видеосервере.
    /// </summary>
    public SetDeviceParametersDelegate SetDeviceParameters;

    /// <summary>
    /// Получение интерфейса ICameraServiceProvider.

```

```

/// </summary>
public GetCameraServiceDelegate GetCameraService;

/// <summary>
/// Получение доп. возможностей (доступные разрешения, фпс, кодеки) конкретного
/// устройства
/// </summary>
public GetAdditionalCapabilitiesDelegate GetAdditionalCapabilities;

/// <summary>
/// Описание портов, которые можно указывать плагину извне
/// </summary>
public List<ExternalNetworkPortDescriptor> ExternalNetworkPortDescriptors;

/// <summary>
/// Список поддерживаемых устройством кодеков при передаче звука
/// </summary>
public OutputSoundCodec SupportedOutputSoundCodec;

/// <summary>
/// Динамическое получение кодека передачи звука. SupportedOutputSoundCodecs
должен
/// быть выставлен в Dynamic.
/// </summary>
public GetOutputSoundCodec GetOutputSoundCodec;

/// <summary>
/// Набор инструментов для автопоиска устройства
/// </summary>
public GetDiscoveryKitDelegate GetDiscoveryKit;

/// <summary>
/// Делегат для смены IP-адреса
/// </summary>
public ChangeIpAddressDelegate ChangeIpAddress;

/// <summary>
/// Требования для смены IP-адреса
/// </summary>
public ChangeIpAddressRequirements ChangeIpAddressRequirements;

/// <summary>
/// Тип архива устройства
/// </summary>
public DeviceStorageType StorageType;

/// <summary>
/// Скорости воспроизведения архива
/// </summary>
public double[] ArchivePlaybackSpeeds;

/// <summary>
/// Список поддерживаемых аналитиков устройства.
/// </summary>
public CameraBuiltInAnalystDescription[] CameraBuiltInAnalystDescriptions;
}

```

Возможности IP-устройства, с которым работает плагин **Получатель кадров**, описываются полем **Capabilities**:

```
/// <summary>
```

```

/// Перечень возможностей устройства
/// </summary>
[Flags]
public enum DevType_Capabilities : ulong
{
    //Коды 512, 1024 и 2048 свободны

    /// <summary>
    /// Устройство работает только с камерами.
    /// </summary>
    SupportsCameras = 1,
    /// <summary>
    /// Устройство поддерживает камеры и видеосерверы.
    /// </summary>
    SupportsCamerasAndServers = 2,
    /// <summary>
    /// Устройство поддерживает работу с альтернативным потоком.
    /// </summary>
    SupportsAlternativeVideoStream = 4,
    /// <summary>
    /// Параметры (разрешение, фпс, компрессия), описываемые массивами
    /// SupportedDeviceParameters/SupportedExtraParameters, не зависят от формата
    /// потока (одинаковы для mjpeg, mpeg4, h264).
    /// </summary>
    /// <remarks>
    /// Данный флаг НАДО ВЫСТАВЛЯТЬ, если медиапуть для выполнения CGI-запроса НЕ
    /// СОДЕРЖИТ "mjpeg", "mpeg4", "h264". В противном случае (если флаг
    используется),
    /// будет всегда дергаться функция SetCameraSettings у класса для подключения
    /// MJPEG, а у других классов она будет игнорироваться (как например на Axis).
    /// </remarks>
    DeviceParametersFormatIndependent = 8,

    /// <summary>
    /// Устройство может иметь архив
    /// </summary>
    SupportsArchive = 16,

    /// <summary>
    /// Устройство поддерживает воспроизведение архива в обратном направлении
    /// </summary>
    DeviceSupportsBackwardDirectionInArchive = 32,

    /// <summary>
    /// Устройство является видеорегистратором
    /// </summary>
    DeviceIsDvr = 64,

    /// <summary>
    /// Необходимо начинать отсчет каналов в регистраторе или видеосервере с одного,
    а
    /// не с нуля как по умолчанию.
    /// </summary>
    ServerStartChannelFromOne = 128,

    /// <summary>
    /// Реализован метод явного получения возможностей устройства
    /// </summary>
    GettingCapabilitiesSupports = 256,

    /// <summary>
    /// Поддержка панорамных камер

```

```

/// </summary>
SupportsPanoramicCameras = 4096,

/// <summary>
/// Поддержка PTZ
/// </summary>
SupportsPtz = 8192,

/// <summary>
/// Поддержка цифровых выходов
/// </summary>
SupportsDigitalOutputs = 16384,

/// <summary>
/// Поддержка омывателя
/// </summary>
SupportsWasherAdjusting = 32768,

/// <summary>
/// Необходимость использования портов из внешних источников
/// </summary>
ExternalNetworkPortsRequired = 65536,

/// <summary>
/// Поддержка коррекции видео с SD-карты
/// </summary>
SupportsSDVideoCorrection = 131072,

/// <summary>
/// Поддержка стабилизации видео с SD-карты
/// </summary>
SupportsSDVideoStabilization = 262144,

/// <summary>
/// Поддержка множественного доступа к архиву устройства
/// </summary>
NotSupportsStorageConcurrentAccess = 524288,

/// <summary>
/// Поддержка домофонов
/// </summary>
SupportsDoorphone = 1048576,

/// <summary>
/// Поддержка работы некоторых функций камеры по защищённому протоколу
/// </summary>
SupportsSecureConnection = 2097152
}

```

Если требуется решить задачу управления поворотной камерой или ее выходами, необходимо реализовать соответствующие интерфейсы **IPtzController** и/или **IDigitalOutputsController**:

```

/// <summary>
/// Унифицированный интерфейс реализации управления поворотной камерой.
/// </summary>
public interface IPtzController
{
    /// <summary>
    /// Инициализация камеры.
    /// </summary>

```

```

/// <returns></returns>
void Reinitialization();

/// <summary>
/// Зачистка ресурсов
/// </summary>
void CleanUp();

/// <summary>
/// Возвращает возможности данной камеры.
/// </summary>
/// <returns></returns>
PtzCapabilities GetCapabilities();

/// <summary>
/// Возвращает названия пресетов, установленных на камере.
/// Количество элементов результирующего массива соответствует количеству пресетов.
/// Каждому пресету соответствует номер, равняющийся индексу в массиве.
/// </summary>
/// <returns>Названия пресетов, установленных на камере.</returns>
string[] GetPresetsNames();

/// <summary>
/// Устанавливает пресет по его номеру.
/// </summary>
/// <param name="presetIndex">Номер пресета.</param>
void SetPresetPosition(int presetIndex);

/// <summary>
/// Устанавливает камеру в "домашнее" положение.
/// </summary>
void MoveToHome();

/// <summary>
/// Прекращает выполнение любой команды PTZ.
/// </summary>
void Stop();

/// <summary>
/// Перемещение на шаг.
/// </summary>
/// <param name="panSpeed">Скорость по горизонтали. Интервал от -100 до 100.</param>
/// <param name="tiltSpeed">Скорость по вертикали. Интервал от -100 до 100.</param>
void StepMove(int panSpeed, int tiltSpeed);

/// <summary>
/// Непрерывное (желательно, плавное) движение.
/// <para/>
/// Если горизонтальная и вертикальная скорость равны 0 - непрерывное движение будет остановлено.
/// </summary>
/// <param name="panSpeed">Скорость по горизонтали. Интервал от -100 до 100.</param>
/// <param name="tiltSpeed">Скорость по вертикали. Интервал от -100 до 100.</param>
void ContiniousMove(int panSpeed, int tiltSpeed);

/// <summary>
/// Относительное приближение.

```

```

/// </summary>
/// <param name="step">Шаг от 1 до 100.</param>
void StepZoomIn(int step);

/// <summary>
/// Относительное отдаление.
/// </summary>
/// <param name="step">Шаг от 1 до 100.</param>
void StepZoomOut(int step);

/// <summary>
/// Непрерывное (желательно плавное) приближение.
/// </summary>
/// <param name="speed">Скорость. Интервал от 1 до 100. Скорость равная 0
означает
/// остановку непрерывного зума.</param>
void ContiniousZoomIn(int speed);

/// <summary>
/// Непрерывное (желательно плавное) отдаление.
/// </summary>
/// <param name="speed">Скорость. Интервал от 1 до 100. Скорость равная 0
означает
/// остановку непрерывного зума.</param>
void ContiniousZoomOut(int speed);

/// <summary>
/// Возвращает максимальное увеличение камеры.
/// </summary>
/// <returns>Максимальное увеличение камеры. Если функция не поддерживается,
/// возвращает отрицательное число.</returns>
double GetMaxZoomFactor();

/// <summary>
/// Возвращает текущее увеличение камеры.
/// </summary>
/// <returns>Текущее увеличение камеры. Если функция не поддерживается,
возвращает
/// отрицательное число.</returns>
double GetCurrentZoomFactor();

/// <summary>
/// Устанавливает абсолютное увеличение. Ничего не делает, если камера это не
/// поддерживает. См. PtzCapabilities.
/// </summary>
void SetZoomFactor(double factor);

/// <summary>
/// Устанавливает максимальное увеличение.
/// </summary>
void ZoomTele();

/// <summary>
/// Устанавливает минимальное увеличение.
/// </summary>
void ZoomWide();

/// <summary>
/// Поворачивает камеру таким образом, что указанная точка оказывается в центре
/// области кадра.
/// </summary>

```



```

/// <param name="point">Точка изображения, которую необходимо поместить в центр
/// путём поворота камеры.
/// Задаётся в пикселях. Начало отсчета(0,0) - левый верхний угол кадра o</param>
/// <param name="frameSize">Размер кадра в пикселях</param>
void MoveTo(System.Drawing.Point point, System.Drawing.Size frameSize);

/// <summary>
/// Поворачивает и масштабирует камеру таким образом,
/// что указанный прямоугольник занимает всю область кадра.
/// Если пропорции прямоугольника не соответствуют пропорциям кадра, то
/// масштабирование производится таким образом, чтобы прямоугольник весь вошёл в
/// кадр.
/// Центр прямоугольника помещается в центр кадра.
/// </summary>
/// <param name="rect">Прямоугольник, задается в пикселях</param>
/// <param name="frameSize">Размер кадра в пикселях</param>
void ShowRect(System.Drawing.Rectangle rect, System.Drawing.Size frameSize);

/// <summary>
/// Устанавливает автоматическое управление фокусом
/// </summary>
void SetAutoFocus();

/// <summary>
/// Дальний фокус
/// </summary>
/// <param name="step">Шаг 1 до 100.</param>
void FocusFar(int step);

/// <summary>
/// Ближний фокус
/// </summary>
/// <param name="step">Шаг 1 до 100.</param>
void FocusNear(int step);

/// <summary>
/// Непрерывный (желательно плавный) дальний фокус
/// </summary>
/// <param name="speed">Скорость. Интервал от 1 до 100. Если скорость равна 0 -
/// непрерывный фокус будет остановлен.</param>
void ContinuousFocusFar(int speed);

/// <summary>
/// Непрерывный (желательно плавный) ближний фокус
/// </summary>
/// <param name="speed">Скорость. Интервал от 1 до 100. Если скорость равна 0
/// - непрерывный фокус будет остановлен.</param>
void ContinuousFocusNear(int speed);

/// <summary>
/// Устанавливает автоматическое управление диафрагмой
/// </summary>
void SetAutoIris();

/// <summary>
/// Закрывает диафрагму.
/// </summary>
/// <param name="step">Шаг 1 до 100.</param>
void IrisOpen(int step);

/// <summary>
/// Приоткрывает диафрагму.

```

```

/// </summary>
/// <param name="step">Шаг 1 до 100.</param>
void IrisClose(int step);

/// <summary>
/// Непрерывное (желательно плавное) открытие диафрагмы.
/// </summary>
/// <param name="speed">Скорость. Интервал от 1 до 100. Если скорость равна 0 -
/// непрерывное открытие диафрагмы будет остановлено.</param>
void ContinuousIrisOpen(int speed);

/// <summary>
/// Непрерывное (желательно плавное) закрытие диафрагмы.
/// </summary>
/// <param name="speed">Скорость. Интервал от 1 до 100. Если скорость равна 0 -
/// непрерывное закрытие диафрагмы будет остановлено.</param>
void ContinuousIrisClose(int speed);

/// <summary>
/// Включить подсветку
/// </summary>
void TurnOnInfraredLight();

/// <summary>
/// Выключить подсветку
/// </summary>
void TurnOffInfraredLight();

/// <summary>
/// Запустить стеклоочиститель
/// </summary>
void TurnOnWiper();

/// <summary>
/// Отстановить стеклоочиститель
/// </summary>
void TurnOffWiper();

/// <summary>
/// Запустить омыватель
/// </summary>
void RunWasher();
}

/// <summary>
/// Унифицированный интерфейс реализации управления выходами камеры
/// </summary>
public interface IDigitalOutputsController
{
    /// <summary>
    /// Инициализация
    /// </summary>
    /// <returns></returns>
    void Reinitialization();

    /// <summary>
    /// Зачистка ресурсов
    /// </summary>
    void CleanUp();

    /// <summary>

```

```

    /// Возвращает возможности данной камеры.
    /// </summary>
    /// <returns></returns>
    DigitalOutputsCapabilities GetCapabilities();

    /// <summary>
    /// Устанавливает заданное значение на выходе
    /// </summary>
    /// <param name="portId">Номер выхода</param>
    /// <param name="value">1 или 0</param>
    void SetOutput(int portId, int value);

    /// <summary>
    /// Выдает последовательность импульсов (ШИМ) на указанном выходе.
    /// Поддерживается не всеми камерами.
    /// </summary>
    /// <param name="portId">Номер выхода</param>
    /// <param name="pulses">Массив импульсов</param>
    void SetOutput(int portId, DigitalImpulse[] pulses);

    /// <summary>
    /// Получить текущие состояния всех выходов
    /// </summary>
    List<DigitalPort> GetOutputsStates();
}

```

Возможности устройства должны возвращаться методом **GetCapabilities()** для информирования хоста о том, какие методы опциональных возможностей (если поддерживаются устройством) реализованы в интерфейсе.

Регистрационную информацию **DevType_RegInfo** устройства необходимо указывать на этапе загрузки сборки с плагином в методе инициализации класса, реализующего интерфейс **IPlugin**, вызывая метод **RegisterDevType** интерфейса хоста (см. [Регистрация плагинов в Macroscop](#)).

Помимо **DevType_RegInfo**, необходимо также заполнить информацию о получателе кадров **MediaStream_RegInfo**:

```

    /// <summary>
    /// Регистрационная информация потока получения данных
    /// </summary>
    public class MediaStream_RegInfo
    {
        /// <summary>
        /// Идентификатор устройства
        /// </summary>
        public Guid DeviceTypeGuid;
        /// <summary>
        /// Формат потока данных
        /// </summary>
        public VideoStreamFormats StreamFormat;
        /// <summary>
        /// Используемый протокол подключения
        /// </summary>
        public NetworkConnectionTypes ConnectionType;

        /// <summary>
        /// Возможности устройства при заданном формате StreamFormat
        /// </summary>
        public MediaStream_Capabilities Capabilities;
    }

```

Данную информацию нужно задать столько раз, сколько различных форматов потока поддерживает IP-устройство.

Аналогично **DevType_RegInfo**, информацию **MediaStream_RegInfo** необходимо задавать

на этапе загрузки сборки с плагином в методе инициализации класса, реализующего интерфейс **IPlugin**. Для этого достаточно вызвать метод **RegisterMediaStreamInfo** интерфейса хоста.

Пример заполнения данных классов доступен в проекте **Camera.csproj**.

Macroscop API с интерфейсами HTTP и RTSP

Macroscop API позволяет обращаться к серверу по HTTP или RTSP интерфейсам для получения видеопотоков реального времени и из архива, а также по HTTP интерфейсу для получения информации о системе и отправки команд системе на выполнение определенных действий. Ниже описаны различные типы запросов.



Пароль пользователя (параметр **password**) передается в виде MD5-хэша в верхнем регистре.

HTTP-интерфейс для получения данных о системе

При комплексном взаимодействии с системой через API и выполнении ряда запросов, перечисленных в данном руководстве, можно получить информацию о конфигурации и текущем состоянии системы, которая не отображается в графическом интерфейсе **Macroscop**. Такая информация может быть получена с помощью CGI-запросов, перечисленных в этом разделе.

Такие запросы в общем виде имеют следующий формат:

```
{Протокол}://{Сервер}:{Порт}/{Ресурс}?login={Логин}&password={Пароль}&{Параметр}={Значение параметра}
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	8080	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https
Ресурс	–	URI ресурса сервера, к которому выполняется запрос
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос. Пользователь должен обладать правами доступа к каналам, функциям и возможностям системы, к которым обращается в рамках запроса
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
Параметр	–	Дополнительный параметр, уточняющий сам запрос или ответ на него. В зависимости от запроса может быть допустимо применение нескольких дополнительных параметров одновременно
Значение параметра	–	Значение применённого дополнительного параметра



В ряде запросов может встречаться одинаковый ресурс **command**. В таком случае тип запроса уточняется параметром **type**.



Некоторые дополнительные параметры обязательны к указанию для выполнения команды.

По умолчанию, данные возвращаются в формате XML. Однако, для части запросов реализована возможность возвращать данные в формате JSON — для этого нужно указать параметр **responsetype=json** или **HTTP-заголовка** "Accept: application/json". Если в описании запроса не указана явным образом возможность возврата данных в JSON, подразумевается, что данные возвращаются только в XML.

Получение конфигурации системы

{...} Доступно получение ответа в формате JSON.

Для выполнения запросов к API системы может потребоваться информация о составляющих действующей конфигурации этой системы. Для получения такой информации используется запрос к ресурсу **configex**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Пример запроса данных в формате **XML**:

```
http://127.0.0.1:8080/configex?login=root&password=
```

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/configex?login=root&password=&responsetype=json
```

Пример ответа на запрос в формате **XML**:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration Id="b4ca3dbb-99a3-47d3-b603-2a631f03a375"
  SenderId="ec3260bd-303d-4c35-8a39-2b80289d3c20"
  Revision="72"
  Timestamp="2023-06-07T06:30:15.3279409Z"
  XMLProtocolVersion="2"
  ServerVersion="4.1.23"
  ProductType="Ultra"
  UseTimeZones="false">
  <Servers>
    <ServerInfo Id="ec3260bd-303d-4c35-8a39-2b80289d3c20"
      Name="Сервер 1"
      Url="192.168.200.87:8080"
      PrimaryIp="192.168.200.87"
      PrimaryPort="8080"
      PrimarySslPort="18080"
      SecondaryIp="192.168.101.223"
      SecondaryPort="8081"
      SecondarySslPort="18081"/>
  </Servers>
  <Channels>
    <ChannelInfo Id="706c4691-3d90-41e3-8789-76eb9810648f"
      Name="Камера 1"
      Description=""
      DeviceInfo="HikVision DS-2xxxxxx, DS-N2xx"
      AttachedToServer="ec3260bd-303d-4c35-8a39-2b80289d3c20"
```

```

        IsDisabled="false"
        IsSoundOn="false"
        IsArchivingEnabled="true"
        IsSoundArchivingEnabled="true"
        AllowedRealtime="true"
        AllowedArchive="true"
        IsPtzOn="false"
        IsTransmitSoundOn="false"
        ArchiveMode="AlwaysOn"
        ArchiveStreamType="Main"
        ArchiveVideoFormat="H264"
        ArchiveRotationMode="None"
        IsFaceAnalystEnabled="false"
        IsPeopleCountingOn="false"
        IsObjectCountingOn="false"
        TimeZoneOffset="5">
<Streams>
  <StreamInfo StreamType="Main"
    StreamFormat="H264"
    RotationMode="None"/>
  <StreamInfo StreamType="Alternative"
    StreamFormat="H264"
    RotationMode="By90ClockwiseDegree"/>
  <StreamInfo StreamType="SecondAlternative"
    StreamFormat="MJPEG"
    RotationMode="By90AntiClockwiseDegree"/>
  <StreamInfo StreamType="ThirdAlternative"
    StreamFormat="MJPEG"
    RotationMode="By180Degree"/>
</Streams>
<UserScenarios>
  <XmlUserScenarioInfo Id="98e9e96b-db9a-4a22-94a8-67c957d7e1fc"
    Name="Ручной запуск тревоги"
    NeedConfirmation="false"/>
</UserScenarios>
<GeoPosition Latitude="53.78760846056143"
  Longitude="-1.5578699111938477"
  Azimuth="90"/>
</ChannelInfo>
</Channels>
<RootSecurityObject Id="f50f5174-1e91-40c2-8d91-ad32119f84f3">
  <ChildSecurityObjects>
    <SecObjectInfo Id="1bee6c88-fdfd-41fd-8d9c-cd8decb8b145" Name="Этаж 1">
      <ChildSecurityObjects/>
      <ChildChannels>
        <ChannelId>a3c5842b-e279-4614-8dc7-9747c5e75899</ChannelId>
      </ChildChannels>
    </SecObjectInfo>
  </ChildSecurityObjects>
</ChildChannels/>
</RootSecurityObject>
<UserGroup>
  <GridTypesAllowed>
    <GridTypes>GridType1</GridTypes>
    <GridTypes>GridType2</GridTypes>
    <GridTypes>GridType3</GridTypes>
    <GridTypes>GridType4</GridTypes>
    <GridTypes>GridType6</GridTypes>
    <GridTypes>GridType7</GridTypes>
    <GridTypes>GridType12X11</GridTypes>
  </GridTypesAllowed>
  <Id>9a8645d1-3665-4b42-b6ef-864fa8f60c64</Id>

```

```

<Name>Администраторы</Name>
<CanConfigure>true</CanConfigure>
<CanConfigureWorkplace>true</CanConfigureWorkplace>
<CanShutdown>true</CanShutdown>
<CanChangeChannelMode>true</CanChangeChannelMode>
<CanManageRec>true</CanManageRec>
<CanAccessExpertMode>true</CanAccessExpertMode>
<CanPTZ>true</CanPTZ>
<PtzPriority>Minimal</PtzPriority>
<CanReceiveSound>true</CanReceiveSound>
<CanTransmitSound>true</CanTransmitSound>
<CanAccessNewCamera>false</CanAccessNewCamera>
<CanAccessReports>true</CanAccessReports>
<CanGetTranscodedVideoFromMobileServer>true</CanGetTranscodedVideoFromMobileServer>
  <CanAccessEditingAnalystPluginsInClient>true</CanAccessEditingAnalystPluginsInClient>
  <CanAccessVideoViaWeb>true</CanAccessVideoViaWeb>
  <CanAccessVideoViaSmartTV>true</CanAccessVideoViaSmartTV>
  <CanExportVideoToAvi>true</CanExportVideoToAvi>
  <CanUseArchiveExport>true</CanUseArchiveExport>
  <CanReceiveMainStream>true</CanReceiveMainStream>
  <IsAllForbidden>false</IsAllForbidden>
  <CanAccessUnifiedLog>true</CanAccessUnifiedLog>
  <CanAccessArchiveMarks>true</CanAccessArchiveMarks>
  <CanAccessSearch>true</CanAccessSearch>
  <CanAccessToAllUsersInUnifiedLog>true</CanAccessToAllUsersInUnifiedLog>
  <CanReceiveMobilePush>true</CanReceiveMobilePush>
  <MessengerCanSendMessage>true</MessengerCanSendMessage>
  <MessengerCanReceiveMessages>true</MessengerCanReceiveMessages>
  <CanConfigureVideowall>true</CanConfigureVideowall>
  <CanBrowsingVideowall>true</CanBrowsingVideowall>
  <CanAccessPlans>true</CanAccessPlans>
  <CanChangePassword>true</CanChangePassword>
  <CanRunUserScenarios>true</CanRunUserScenarios>
  <CanAccessGis>true</CanAccessGis>
</UserGroup>
<MobileServerInfo IsEnabled="true"
  IsProxyEnabled="true"
  IsMobilePushEnabled="true"
  Port="8089"
  UsePFrames="false"
  FpsLimit="0"
  LowResolution="120 x 90"
  MiddleResolution="240 x 180"
  HighResolution="800 x 480">
  <Resolutions>
    <ResolutionInfo Width="800"
      Height="480"
      IsEnabled="true"
      FpsLimit="15"
      UsePFrames="true"
      Type="High"/>
    <ResolutionInfo Width="240"
      Height="180"
      IsEnabled="true"
      FpsLimit="4"
      UsePFrames="false"
      Type="Middle"/>
    <ResolutionInfo Width="120"
      Height="90"
      IsEnabled="false"

```



```

        FpsLimit="4"
        UsePframes="false"
        Type="Low"/>
    </Resolutions>
</MobileServerInfo>
<RtspServerInfo IsEnabled="true"
    TcpPort="554"
    IsMjpegEnabled="true"/>
<MobileDevicesCapabilities>
    <Archive>true</Archive>
    <Ptz>true</Ptz>
    <Hls>true</Hls>
    <AppleMobilePush>true</AppleMobilePush>
    <AndroidMobilePush>true</AndroidMobilePush>
    <Profiles>true</Profiles>
    <UserScenarios>true</UserScenarios>
    <SmartAssistant>true</SmartAssistant>
    <Gis>true</Gis>
</MobileDevicesCapabilities>
<WorldMapConfig>
    <Locations Name="Все локации">
        <ChildLocations>
            <Location Name="Местный офис"
                Latitude="53.788007776832465"
                Longitude="-1.541484296321869"
                Zoom="19"/>
            <Location Name="Головной офис"
                Latitude="53.787638567873074"
                Longitude="-1.543552279472351"
                Zoom="19"/>
        </ChildLocations>
        <ChildFolders>
            <FolderLocation Name="Строящиеся локации">
                <ChildLocations>
                    <Location Name="Бизнес-центр"
                        Latitude="53.788853934465465"
                        Longitude="-1.5443569421768188"
                        Zoom="19"/>
                </ChildLocations>
            </FolderLocation>
        </ChildFolders>
    </Locations>
</WorldMapConfig>
</Configuration>

```

Пример ответа на запрос в формате **JSON**:

```

{
  "Id": "b4ca3dbb-99a3-47d3-b603-2a631f03a375",
  "SenderId": "ec3260bd-303d-4c35-8a39-2b80289d3c20",
  "RevNum": 72,
  "Timestamp": "2023-06-07T06:30:15.3279409Z",
  "XmlProtocolVersion": 2,
  "ServerVersion": "4.1.23",
  "ProductType": "Ultra",
  "Servers": [
    {
      "Id": "ec3260bd-303d-4c35-8a39-2b80289d3c20",
      "Name": "Сервер 1",
      "Url": "192.168.200.87:8080",

```

```

    "PrimaryIp": "192.168.200.87",
    "PrimaryPort": "8080",
    "PrimarySslPort": "18080",
    "SecondaryIp": "",
    "SecondaryPort": "0",
    "SecondarySslPort": "0",
    "ConnectionUrl": null
  }
],
"Channels": [
  {
    "Id": "706c4691-3d90-41e3-8789-76eb9810648f",
    "Name": "Камера 1",
    "Description": "",
    "DeviceInfo": "HikVision DS-2xxxxxx, DS-N2xx",
    "AttachedToServer": "ec3260bd-303d-4c35-8a39-2b80289d3c20",
    "IsDisabled": false,
    "IsSoundOn": false,
    "IsArchivingEnabled": true,
    "IsSoundArchivingEnabled": true,
    "AllowedRealtime": true,
    "AllowedArchive": true,
    "IsPtzOn": false,
    "IsTransmitSoundOn": false,
    "ArchiveMode": "AlwaysOn",
    "Streams": [
      {
        "StreamType": "Main",
        "StreamFormat": "H264",
        "RotationMode": "None"
      },
      {
        "StreamType": "Alternative",
        "StreamFormat": "H264",
        "RotationMode": "By90ClockwiseDegree"
      },
      {
        "StreamType": "SecondAlternative",
        "StreamFormat": "MJPEG",
        "RotationMode": "By90AntiClockwiseDegree"
      },
      {
        "StreamType": "ThirdAlternative",
        "StreamFormat": "MJPEG",
        "RotationMode": "By180Degree"
      }
    ],
    "UserScenarios": [
      {
        "Id": "98e9e96b-db9a-4a22-94a8-67c957d7e1fc",
        "Name": "Ручной запуск тревоги",
        "NeedConfirmation": false
      }
    ],
    "ArchiveStreamType": "Main",
    "ArchiveVideoFormat": "H264",
    "ArchiveRotationMode": "None",
    "IsFaceRecOn": false,
    "GeoPosition": {
      "Latitude": 53.78760846056143,
      "Longitude": -1.5578699111938477,
      "Azimuth": 90.0
    }
  }
]

```

```

    },
    "IsPeopleCountingOn": false,
    "IsObjectCountingOn": false,
    "TimeZoneOffset": 5.0
  }
],
"RootSecObject": {
  "ChildSecObjects": [
    {
      "ChildSecObjects": [],
      "ChildChannels": [
        "a3c5842b-e279-4614-8dc7-9747c5e75899"
      ],
      "Id": "1bee6c88-fdfd-41fd-8d9c-cd8decb8b145",
      "Name": "Этаж 1"
    }
  ],
  "ChildChannels": [],
  "Id": "f50f5174-1e91-40c2-8d91-ad32119f84f3",
  "Name": null
},
"UserGroup": {
  "GridTypesAllowed": [
    "GridType1",
    "GridType2",
    "GridType3",
    "GridType4",
    "GridType6",
    "GridType7",
    "GridType12X11"
  ],
  "Id": "9a8645d1-3665-4b42-b6ef-864fa8f60c64",
  "Comment": null,
  "Name": "Администраторы",
  "CanConfigure": true,
  "CanConfigureWorkplace": true,
  "CanShutdown": true,
  "CanChangeChannelMode": true,
  "CanManageRec": true,
  "CanAccessExpertMode": true,
  "CanPTZ": true,
  "PtzPriority": "Minimal",
  "CanReceiveSound": true,
  "CanTransmitSound": true,
  "CanAccessNewCamera": false,
  "CanAccessReports": true,
  "CanGetTranscodedVideoFromMobileServer": true,
  "CanAccessEditingAnalystPluginsInClient": true,
  "CanAccessVideoViaWeb": true,
  "CanAccessVideoViaSmartTV": true,
  "CanExportVideoToAvi": true,
  "CanUseArchiveExport": true,
  "CanReceiveMainStream": true,
  "IsAllForbidden": false,
  "CanAccessUnifiedLog": true,
  "CanAccessArchiveMarks": true,
  "CanAccessSearch": true,
  "CanAccessToAllUsersInUnifiedLog": true,
  "CanReceiveMobilePush": true,
  "MessengerCanSendMessages": true,
  "MessengerCanReceiveMessages": true,
  "CanConfigureVideowall": true,

```

```

    "CanBrowsingVideowall": true,
    "CanAccessPlans": true,
    "CanChangePassword": true,
    "CanRunUserScenarios": true,
    "CanAccessGis": true
  },
  "MobileServerInfo": {
    "IsEnabled": true,
    "IsProxyEnabled": true,
    "IsMobilePushEnabled": true,
    "Port": 8089,
    "UsePFrames": false,
    "FpsLimit": 0,
    "LowResolution": "120 x 90",
    "MiddleResolution": "240 x 180",
    "HighResolution": "800 x 480",
    "Resolutions": [
      {
        "Width": 800,
        "Height": 480,
        "IsEnabled": true,
        "FpsLimit": 15,
        "UsePFrames": true,
        "Type": "High"
      },
      {
        "Width": 240,
        "Height": 180,
        "IsEnabled": true,
        "FpsLimit": 4,
        "UsePFrames": false,
        "Type": "Middle"
      },
      {
        "Width": 120,
        "Height": 90,
        "IsEnabled": false,
        "FpsLimit": 4,
        "UsePFrames": false,
        "Type": "Low"
      }
    ]
  },
  "RtspServerInfo": {
    "IsEnabled": true,
    "TcpPort": 554,
    "IsMjpegEnabled": true
  },
  "MobileDevicesCapabilities": {
    "Archive": true,
    "Ptz": true,
    "Hls": true,
    "AppleMobilePush": true,
    "AndroidMobilePush": true,
    "Profiles": true,
    "UserScenarios": true,
    "SmartAssistant": true,
    "Gis": true
  },
  "WorldMapConfig": {
    "Locations": {
      "Name": "Все локации",

```

```

    "ChildLocations": [
      {
        "Name": "Местный офис",
        "Latitude": 53.788007776832465,
        "Longitude": -1.541484296321869,
        "Zoom": 19.0
      },
      {
        "Name": "Головной офис",
        "Latitude": 53.787638567873074,
        "Longitude": -1.543552279472351,
        "Zoom": 19.0
      }
    ],
    "ChildFolders": [
      {
        "Name": "Строящиеся локации",
        "ChildLocations": [
          {
            "Name": "Бизнес-центр",
            "Latitude": 53.788853934465465,
            "Longitude": -1.5443569421768188,
            "Zoom": 19.0
          }
        ]
      },
      "ChildFolders": []
    ]
  ],
  "UseTimeZones": false
}

```

Независимо от выбранного способа представления данных, ответ от сервера будет содержать подробную информацию о действующей конфигурации системы, разделённую на несколько секций.

Секция **Configuration**, помимо структурного включения остальных секций ответа, содержит следующие собственные элементы:

Параметр	Описание
Id	Уникальный идентификатор действующей конфигурации
SenderId или RevNum	Уникальный идентификатор сервера, отправившего ответ
Revision	Номер ревизии действующей конфигурации. Номер увеличивается на единицу после каждого применения изменений в конфигурации
Timestamp	Временная метка последнего применения конфигурации
XMLProtocolVersion	Версия протокола XML. На данный момент номер протокола равняется 2, его смена в будущем предполагает возможность появления новых элементов и атрибутов в xml-ответе
ServerVersion	Версия Macroscop , установленная на сервере, отправившем ответ
ProductType	Тип лицензии, активированной на сервере, отправившем ответ
UseTimeZones	Состояние опции Учитывать часовые пояса . Возможные значения: true — опция включена, false — выключена

Секция **Servers** содержит описание серверов, входящих в действующую конфигурацию. Каждый сервер описывается элементом **ServerInfo**, в который входят следующие элементы:

Параметр	Описание
Id	Уникальный идентификатор сервера
Name	Название сервера в рамках действующей конфигурации
Url	URL сервера
PrimaryIp	Основной IP адрес сервера
PrimaryPort	Основной HTTP порт сервера
PrimarySslPort	Основной HTTPS порт сервера
SecondaryIp	Дополнительный IP адрес сервера
SecondaryPort	Дополнительный HTTP порт сервера
SecondarySslPort	Дополнительный HTTPS порт сервера

Секция **Channels** содержит описание настроек каналов действующей конфигурации. Настройки каждого канала описываются элементом **ChannelInfo**, который состоит из следующих атрибутов и вложенных секций:

Параметр	Описание
Id	Уникальный идентификатор канала. Может быть использован в других запросах для указания нужной камеры параметром channelid
Name	Имя канала в рамках действующей конфигурации
Description	Описание камеры. Настраивается с помощью REST API
DeviceInfo	Заданные каналу Производитель и Модель устройства
AttachedToServer	Уникальный идентификатор сервера, к которому прикреплен канал
IsDisabled	Состояние канала в действующей конфигурации. Возможные значения: true — канал выключен, false — включен
IsSoundOn	Состояние опции получения звука от камеры на данном канале. Если этот параметр равен false , то параметр IsSoundArchivingEnabled может быть проигнорирован. Возможные значения: true — получение звука включено, false — выключено
IsArchivingEnabled	Состояние опции архивирования видеоданных для канала. Возможные значения: true — архивирование включено, false — выключено
IsSoundArchivingEnabled	Состояние опции архивирования звуковых данных для канала. Возможные значения: true — архивирование включено, false — выключено
AllowedRealtime	Доступность просмотра видео от данного канала в реальном времени для пользователя, отправившего запрос. Возможные значения: true — просмотр разрешён, false — запрещён
AllowedArchive	Доступность просмотра видео от данного канала из архива для пользователя, отправившего запрос. Возможные значения: true — просмотр разрешён, false — запрещён
IsPtzOn	Состояние опции управления PTZ камерой в рамках данного канала. Возможные значения: true — управление включено, false — выключено

IsTransmitSoundOn	Состояние опции передачи звука на динамик камеры. Возможные значения: true — передача звука включена, false — выключена
ArchiveMode	Режим записи архива, заданный для данного канала. Возможные значения: AlwaysOn — запись активна всегда, OnlyManual — активируется вручную, BySchedule — активируется по расписанию, MDandManual — активируется вручную и автоматически по команде от детектора движения
ArchiveStreamType	Поток видео, заданный для записи архива. Возможные значения идентичны элементу StreamType
ArchiveVideoFormat	Заданный формат для архивируемого потока видео. Возможные значения идентичны элементу StreamFormat
ArchiveRotationMode	Заданные настройки поворота изображения для архивируемого потока видео. Возможные значения идентичны элементу RotationMode
IsFaceAnalystEnabled или IsFaceRecOn	Состояние модуля Распознавание лиц. Возможные значения: true — модуль включен, false — выключен
IsPeopleCountingOn	Состояние модуля Подсчет посетителей. Возможные значения: true — модуль включен, false — выключен
IsObjectCountingOn	Состояние модуля Подсчет объектов. Возможные значения: true — модуль включен, false — выключен
TimeZoneOffset	Часовой пояс камеры относительно UTC
Streams	<p>Вложенная секция, содержащая настройки потоков видеоданных канала. Количество подсекций зависит от настроек получения потоков для данного канала. Содержит следующие элементы:</p> <ul style="list-style-type: none"> • StreamType — тип потока. Возможные значения: Main — Основной, Alternative — Дополнительный 1, SecondAlternative — Дополнительный 2, ThirdAlternative — Дополнительный 3. • StreamFormat — формат потока видео. Возможные значения: H264, Hvc (соответствует H.265), MJPEG, MPEG4_Part2, MxPeg. • RotationMode — настройки поворота изображения для потока видео. Возможные значения: None — без поворота, By90ClockwiseDegree — 90 градусов по часовой стрелке, By90AntiClockwiseDegree — 90 градусов против часовой стрелки, By180Degree — 180 градусов
UserScenarios	<p>Вложенная секция, содержащая информацию о созданных для канала Пользовательских задачах. Содержит следующие элементы:</p> <ul style="list-style-type: none"> • Id — уникальный идентификатор задачи. • Name — имя задачи в рамках действующей конфигурации. • NeedConfirmation — состояние опции Подтверждение действия. Возможные значения: true — задача требует подтверждение выполнения, false — задача запускается без дополнительного подтверждения
GeoPosition	<p>Вложенная секция, содержащая информацию о координатах камеры на карте. Доступна только для систем с лицензиями Enterprise и Ultra. Содержит следующие элементы:</p> <ul style="list-style-type: none"> • Latitude — широта. • Longitude — долгота. • Azimuth — азимут

Секция **RootSecurityObject** содержит информацию о структуре дерева объектов безопасности и принадлежности к ним каналов. Состоит из следующих элементов:

Параметр	Описание
Id	Уникальный идентификатор объекта безопасности
Name	Имя объекта в рамках действующей конфигурации
ChildChannels	Идентификаторы каналов, существующих в рамках объекта
ChildSecurityObjects	Вложенная секция, соответствующая объекту, вложенному в другой объект (Подкаталог). В свою очередь содержит список элементов, идентичный корневой секции RootSecurityObject

Секция **UserGroup** содержит информацию о настройках группы, к которой принадлежит пользователь, выполнивший запрос конфигурации. Состоит из следующих элементов и вложенных секций:

Параметр	Описание
Id	Уникальный идентификатор группы
Comment	Комментарий, добавленный группе в рамках действующей конфигурации
Name	Имя группы в рамках действующей конфигурации
GridTypesAllowed	Список сеток, доступных данной группе для создания видов
—	Список прав доступа к различным элементам системы. Возможные значения для прав: true — право включено, false — право отключено

В секции **MobileServerInfo** содержится информация о параметрах перекодирования видео мобильным сервером. Состоит из следующих элементов и вложенных секций:

Параметр	Описание
IsEnabled	Состояние механизма перекодирования потоков мобильным сервером. Возможные значения: true — перекодирование потоков в указанные параметры включено, false — выключено
IsProxyEnabled	Перенаправление запросов на HTTP(S) порт вместо использования собственного порта мобильного сервера. Возможные значения: true — используется HTTP(S) порт сервера, false — используется собственный порт мобильного сервера
IsMobilePushEnabled	Опция отправки Push уведомлений мобильным приложениям. Возможные значения: true — отправка Push уведомлений разрешена, false — отправка запрещена
Port	Собственный порт мобильного сервера. Используется для взаимодействия с мобильными приложениями, если параметр IsProxyEnabled=false
UsePFFrames	Использование разностных кадров (P-кадров) при передаче потоков мобильным приложениям. Возможные значения: true — передаются как опорные (I-), так и разностные (P-) кадры, false — передаются только опорные кадры
FpsLimit	Ограничение количества кадров в секунду при перекодировании потока
LowResolution	Информация о заданном разрешении потока Плохого качества
MiddleResolution	Информация о заданном разрешении потока Среднего качества

HighResolution	Информация о заданном разрешении потока Хорошего качества
Resolutions	Вложенная секция, содержащая детальную информацию о настройках потоков Плохого , Среднего и Хорошего качества. Состоит из следующих элементов: <ul style="list-style-type: none"> • Width — ширина кадра. • Height — высота кадра. • Type — тип потока. Возможные значения: High, Middle, Low.

Секция **RtspServerInfo** содержит настройки RTSP сервера, используемого для получения потоков видео из системы сторонними приложениями. Состоит из следующих элементов:

Параметр	Описание
IsEnabled	Доступность RTSP сервера для подключения. Возможные значения: true — включено, false — выключено
TcpPort	Порт для подключения к RTSP серверу
IsMjpegEnabled	Состояние опции Разрешить вещание Mjpeg по протоколу RTSP . Возможные значения: true — включено, false — выключено

Секция **MobileDevicesCapabilities** содержит информацию о возможностях взаимодействия мобильных приложений с системой.



Содержимое этой секции определяется версией системы, а не её настройками. Возможность использования той или иной опции зависит от наличия её поддержки в установленной версии **Macroscop**. Значение элементов в данной секции всегда **true**.

Состоит из следующих элементов:

Параметр	Описание
Archive	Возможность просматривать архив системы с помощью мобильного приложения
Ptz	Возможность управлять PTZ функционалом камер с помощью мобильного приложения
Hls	Поддержка протокола HLS со стороны сервера
AppleMobilePush	Возможность сервера отправлять Push-уведомления мобильным приложениям на iOS
AndroidMobilePush	Возможность сервера отправлять Push-уведомления мобильным приложениям на Android
Profiles	Доступность для мобильных приложений созданных в системе видов
UserScenarios	Возможность инициировать выполнение пользовательских сценариев, настроенных для камер системы, из мобильных приложений
SmartAssistant	Возможность взаимодействия с системой с помощью ассистента EVA
Gis	Доступ мобильных приложений к картам и размещённым на них объектам

Секция **WorldMapConfig** содержит настройки Локаций, созданных на картах. Доступно только для систем с лицензиями **Enterprise** и **Ultra**. Состоит из следующих элементов и вложенных секций:

Параметр	Описание
Name	Имя локации или группы локаций
Latitude	Значение широты для центральной точки локации
Longitude	Значение долготы для центральной точки локации
Zoom	Степень увеличения карты для данной локации
ChildLocations	Вложенная секция, содержащая список локаций, входящих в группу
ChildFolders	Вложенная секция, содержащая группу локаций, созданную в рамках другой группы локаций (Подкаталог). В свою очередь содержит список элементов, идентичный корневой секции WorldMapConfig

Получение списка доступных сеток на клиенте Macroscop

{...} Доступно получение ответа в формате JSON.

Для получения списка сеток, которые может использовать пользователь, от чьего имени в настоящий момент выполнено подключение клиентского приложения к серверу **Macroscop**, используется запрос к ресурсу **command** с типом **getgrids**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение. Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения (Отсчёт с 0). Обязательный параметр
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Пример запроса данных в формате **XML**:

```
http://127.0.0.1:8080/command?type=getgrids&login=root&password=&clientip=192.168.1.2&monitor=0
```

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/command?type=getgrids&login=root&password=&clientip=192.168.1.2&monitor=0&responsetype=json
```

В ответ сервер возвращает список доступных сеток на указанном клиенте.

Пример ответа на запрос в формате **xml**:

```
<ArrayOfString>
  <string>1</string>
```

```

<string>4</string>
<string>6_1</string>
<string>7</string>
<string>8_1</string>
<string>9</string>
<string>10</string>
<string>13</string>
<string>16</string>
<string>25</string>
</ArrayOfString>

```

Пример ответа на запрос в формате **json**:

```

[
  "1",
  "4",
  "6_1",
  "7",
  "8_1",
  "9",
  "10",
  "13",
  "16",
  "25"
]

```

Где:

Параметр	Описание
Значение до «_»	Число ячеек в сетке
Значение после «_»	Номер конфигурации



Номер конфигурации введен только для сеток, обладающих одинаковым числом ячеек, но отличающихся размером и расположением ячеек.

Полученные данные можно использовать для [смены сетки на клиенте](#).

Получение списка серверных профилей экрана (видов)

{...} Доступно получение ответа в формате JSON.

Для получения списка профилей экрана (более известных как виды), созданных на сервере, используется запрос к ресурсу **command** с типом **getprofiles**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Пример запроса данных в формате **XML**:

```
http://127.0.0.1:8080/command?type=getprofiles&login=root&password=
```

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/command?type=getprofiles&login=root&password=&responsetype=json
```

В ответ сервер возвращает список серверных профилей, доступных для пользователя, отправившего запрос.



Вид считается недоступным если не содержит камер, доступных пользователю для просмотра, а также при отсутствии у пользователя прав на использование этого вида.

Пример ответа на запрос в формате **XML**:

```
<ArrayOfMapInfo>
  <MapInfo>
    <Id>22ab7b5e-a8e0-48f9-80ff-a48217baa21f</Id>
    <Name>Офис</Name>
  </MapInfo>
  <MapInfo>
    <Id>f2294ff9-783c-4b49-b5ab-a8fd0529da99</Id>
    <Name>Склад</Name>
  </MapInfo>
</ArrayOfMapInfo>
```

Пример ответа на запрос в формате **JSON**:

```
[
  {
    "Id": "22ab7b5e-a8e0-48f9-80ff-a48217baa21f",
    "Name": "Офис"
  },
  {
    "Id": "f2294ff9-783c-4b49-b5ab-a8fd0529da99",
    "Name": "Склад"
  }
]
```

Где:

Параметр	Описание
Id	Уникальный идентификатор профиля. Может быть использован при отправке запроса на установку профиля на клиенте
Name	Имя профиля в рамках действующей конфигурации

Получение информации о текущем профиле экрана в клиенте Macroscop

{...} Доступно получение ответа в формате JSON.

Для получения информации о профиле, используемом на мониторе выбранного клиента **Macroscop** в настоящий момент, используется запрос к ресурсу **command** с типом **getcurrentgrid**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0. Обязательный параметр
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json



Значение **clientip** должно содержать фактический адрес клиента **Macroscop**. Использование localhost (127.0.0.1) вызовет ошибку даже при обращении к клиенту, установленному на сервере.

Пример запроса данных в формате **XML**:

```
http://127.0.0.1:8080/command?type=getcurrentgrid&login=root&password=&clientip=192.168.1.2&monitor=0
```

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/command?type=getcurrentgrid&login=root&password=&clientip=192.168.1.2&monitor=0&responsetype=json
```

В ответ сервер возвращает информацию о текущем профиле, типе использованной в нём сетки и его содержимом.

Пример ответа на запрос в формате **XML**:

```
<ViewInfo>
  <GridType>GridType3</GridType>
  <ViewInfoId>20db65ef-17fb-4d75-9513-acb8535d552b</ViewInfoId>
  <GridCells>
    <GridCellParameters>
      <ChannelId>a3785d7f-9259-456b-a64a-048342f22964</ChannelId>
      <CellIndex>0</CellIndex>
      <IsArchive>>false</IsArchive>
      <ArchiveStartTime xsi:nil="true"/>
      <ArchivePlaySpeed>1</ArchivePlaySpeed>
    </GridCellParameters>
    <GridCellParameters>
```

```

    <ChannelId>706c4691-3d90-41e3-8789-76eb9810648f</ChannelId>
    <CellIndex>1</CellIndex>
    <IsArchive>>true</IsArchive>
    <ArchiveStartTime>2023-06-09T14:09:43.7434941Z</ArchiveStartTime>
    <ArchivePlaySpeed>1</ArchivePlaySpeed>
  </GridCellParameters>
  <GridCellParameters>
    <ChannelId>00000000-0000-0000-0000-000000000000</ChannelId>
    <CellIndex>2</CellIndex>
    <IsArchive>>false</IsArchive>
    <ArchiveStartTime xsi:nil="true"/>
    <ArchivePlaySpeed>0</ArchivePlaySpeed>
  </GridCellParameters>
</GridCells>
<ViewName>Офис</ViewName>
</ViewInfo>

```

Пример ответа на запрос в формате **JSON**:

```

{
  "GridType": "GridType3",
  "ViewInfoId": "20db65ef-17fb-4d75-9513-acb8535d552b",
  "GridCells": [
    {
      "ChannelId": "a3785d7f-9259-456b-a64a-048342f22964",
      "CellIndex": 0,
      "IsArchive": false,
      "ArchiveStartTime": null,
      "ArchivePlaySpeed": 1.0
    },
    {
      "ChannelId": "706c4691-3d90-41e3-8789-76eb9810648f",
      "CellIndex": 1,
      "IsArchive": true,
      "ArchiveStartTime": "2023-06-09T14:09:43.7434941Z",
      "ArchivePlaySpeed": 1.0
    },
    {
      "ChannelId": "00000000-0000-0000-0000-000000000000",
      "CellIndex": 2,
      "IsArchive": false,
      "ArchiveStartTime": null,
      "ArchivePlaySpeed": 0.0
    }
  ],
  "ViewName": "Офис"
}

```

Где:

Параметр	Описание
GridType	Тип сетки, использованной для выбранного профиля
ViewInfoId	Уникальный идентификатор профиля

GridCells	<p>Вложенная секция, описывающая содержимое ячеек выбранного профиля.</p> <ul style="list-style-type: none"> • ChannelId – уникальный идентификатор канала, размещённого в ячейке. • CellIndex – номер ячейки в виде (Отсчёт с нуля). • IsArchive – состояние режима отображения архива в ячейке. Возможные значения: true – ячейка находится в режиме просмотра архива, false – отображение потока реального времени. • ArchiveStartTime – текущая позиция в архиве в UTC. • ArchivePlaySpeed – скорость воспроизведения видео в ячейке. Возможные значения: 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 60, 120. При отображении потока реального времени значение всегда 1
ViewName	Имя профиля в рамках действующей конфигурации



Полученные данные можно использовать при [установке профиля на клиенте](#).

Получение текущего времени сервера Macroscop



Доступно получение ответа в формате JSON.

Для получения информации о текущем времени компьютера, на котором работает сервер **Macroscop**, используется запрос к ресурсу **command** с типом **gettime**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml, json

Пример запроса данных в формате **XML**:

```
http://127.0.0.1:8080/command?type=gettime&login=root&password=
```

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/command?type=gettime&login=root&password=&responsetype=json
```

В ответ сервер возвращает текущее время компьютера в часовом поясе UTC.

Пример ответа на запрос в формате **XML**:

```
<string>14.06.2023 07:01:30</string>
```

Пример ответа на запрос в формате **JSON**:

```
"14.06.2023 07:01:30"
```

Получение информации о наличии архива в указанный момент времени

{...} Доступно получение ответа в формате JSON.

Для получения информации о наличии данных в архиве за определённый момент времени используется запрос к ресурсу **command** с типом **findarchive**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
searchTime	–	Дата и время, для которых необходимо найти архивные записи. Время должно указываться в часовом поясе UTC. Обязательный параметр
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json



Параметр **searchTime** поддерживает несколько форматов записи даты и времени:

- Дата может записываться как с полным обозначением года («YYYY» или «2023»), так и с сокращённым («YY» или «23»).
- Для разделения числа, месяца и года в дате можно использовать как точку («.»), так и дефис («-»).
- Допускается написание даты, начиная как с числа («DD.ММ.YYYY» или «14.06.2023»), так и с года («YYYY.ММ.DD» или «2023.06.14»).
- Разделение даты и времени может осуществляться как пробелом (« ») или его кодом («%20»), так и плюсом («+»).



Несмотря на то, что ответ от сервера содержит разделитель «Т» между датой и временем, этот разделитель не поддерживается в теле запроса.

Пример запроса данных в формате **XML**:

```
http://127.0.0.1:8080/command?type=findarchive&login=root&password=&channelid=a3785d7f-9259-456b-a64a-048342f22964&searchTime=14.06.2023+10:10:10
```

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/command?type=findarchive&login=root&password=&channelid=a3785d7f-9259-456b-a64a-048342f22964&searchTime=14.06.2023+10:10:10&responsetype=json
```

В ответ сервер возвращает сведения о наличии архива за указанное время и временную метку ближайшего к заданному времени видеокadra.

Пример ответа на запрос в формате **XML**:

```
<CheckArchiveResult>
  <HasArchive>true</HasArchive>
  <NearestFrameTimestamp>2023-06-14T10:10:00.0988432Z</NearestFrameTimestamp>
</CheckArchiveResult>
```


Пример ответа на запрос в формате **JSON**:

```
{
  "HasArchive": true,
  "NearestFrameTimestamp": "2023-06-14T10:10:00.0988432Z"
}
```

Где:

Параметр	Описание
HasArchive	Флаг наличия архива за указанное время. Возможные значения: true – архив за указанное время существует, false – не существует
NearestFrameTimestamp	Временная метка ближайшего видеокadra в формате UTC. Если HasArchive=false , параметр примет значение null

Получение списка интервалов с информацией о начале и окончании записи архива

{..} Доступно получение ответа в формате JSON.

Для получения информации о наличии в архиве интервалов записи за указанный период используется запрос к ресурсу **archivefragments**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
fromtime	–	Дата и время начала периода, за который запрашивается информация. Время должно указываться в часовом поясе UTC. Обязательный параметр.
totime	–	Дата и время конца периода, за который запрашивается информация. Время должно указываться в часовом поясе UTC. Обязательный параметр
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml, json



При указании параметров **fromtime** и **totime** необходимо соблюдать следующие требования:

- Дата должна быть записана с полным указанием года («2023», но не «23»).
- В качестве разделителя в дате может использоваться только точка («.»).
- Порядок написания даты должен быть DD.MM.YYYY («14.06.2023»).
- Разделение даты и времени может осуществляться как пробелом (« ») или его кодом («%20»), так и плюсом («+»).



Несмотря на то, что ответ от сервера содержит иной формат записи даты и разделитель «Т» между датой и временем, они не поддерживаются в теле запроса.

Пример запроса данных в формате **xml**:

```
http://127.0.0.1:8080/archivefragments?login=root&password=&channelid=a3785d7f-9259-456b-a64a-048342f22964&fromtime=14.06.2023+00:00:00&totime=14.06.2023+10:00:00
```

Пример запроса данных в формате **json**:

```
http://127.0.0.1:8080/archivefragments?login=root&password=&channelid=a3785d7f-9259-456b-a64a-048342f22964&fromtime=14.06.2023+00:00:00&totime=14.06.2023+10:00:00&responsetype=json
```

В ответ сервер вернёт массив данных, элементы которого содержат время начала и окончания записи архива, например, когда запись велась по детектору движения.

Пример ответа на запрос в формате **XML**:

```
<ArchiveFragmentsList>
  <Fragments>
    <ArchiveFragment>
      <Id>50b4d1f2-10ea-4511-940c-136217f2c84f</Id>
      <FromTime>2023-06-14T02:11:25.880943Z</FromTime>
      <ToTime>2023-06-14T09:13:22.7468942Z</ToTime>
    </ArchiveFragment>
    <ArchiveFragment>
      <Id>8c92e69f-2d17-4b39-a3bf-e43e4d93bc71</Id>
      <FromTime>2023-06-14T09:20:12.880447Z</FromTime>
      <ToTime>2023-06-14T14:09:43.7434942Z</ToTime>
    </ArchiveFragment>
  </Fragments>
</ArchiveFragmentsList>
```

Пример ответа на запрос в формате **JSON**:

```
{
  "Fragments": [
    {
      "Id": "50b4d1f2-10ea-4511-940c-136217f2c84f ",
      "FromTime": "2023-06-14T02:11:25.880943Z ",
      "ToTime": "2023-06-14T09:13:22.7468942Z "
    },
    {
      "Id": "8c92e69f-2d17-4b39-a3bf-e43e4d93bc71",
      "FromTime": "2023-06-14T09:20:12.880447Z ",
      "ToTime": "2023-06-14T14:09:43.7434942Z "
    }
  ]
}
```

Где:

Параметр	Описание
Id	Уникальный идентификатор фрагмента архива
FromTime	Время начала фрагмента архива
ToTime	Время окончания фрагмента архива



Количество отдельных фрагментов архива за заданный период определяется непрерывностью записи архива для указанного канала. Если архив ведётся непрерывно, ответ может содержать лишь один фрагмент архива даже при условии запроса данных за несколько дней.

Получение информации о состоянии каналов



Доступно получение ответа в формате JSON.

Для получения информации о состоянии каналов и их опций используется запрос к ресурсу **command** с типом **getchannelsstates**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Пример запроса данных в формате **XML**:

```
http://127.0.0.1:8080/command?type=getchannelsstates&login=root&password=
```

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/command?type=getchannelsstates&login=root&password=&responsetype=json
```

В ответ сервер передаст информацию о состоянии потоков данных, использующихся активными каналами.



Ответ на запрос содержит информацию только о включенных каналах. Информация о существующих, но выключенных каналах не передаётся.

Пример ответа на запрос в формате **XML**:

```
<ArrayOfChannelState>
  <ChannelState>
    <Id>a3785d7f-9259-456b-a64a-048342f22964</Id>
    <IsRecordingOn>true</IsRecordingOn>
    <StreamsStates>
      <Stream>
        <Type>MainVideo</Type>
```

```

    <State>NoConnection</State>
  </Stream>
  <Stream>
    <Type>AlternativeVideo</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>MainSound</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>AlternativeSound</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>OutputSound</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>MotionDetection</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>IO</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>ArchiveVideo</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>ArchiveSound</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>SecondAlternativeVideo</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>ThirdAlternativeVideo</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>SecondAlternativeSound</Type>
    <State>Stopped</State>
  </Stream>
  <Stream>
    <Type>ThirdAlternativeSound</Type>
    <State>Stopped</State>
  </Stream>

```

```

    </Stream>
  </StreamsStates>
  <IsAnalogCameraConnected>>false</IsAnalogCameraConnected>
</ChannelState>
</ArrayOfChannelState>

```

Пример ответа на запрос в формате **JSON**:

```

[
  {
    "Id": "a3785d7f-9259-456b-a64a-048342f22964",
    "IsRecordingOn": false,
    "StreamsStates":
    [
      {
        "Type": "MainVideo",
        "State": "Active"
      },
      {
        "Type": "AlternativeVideo",
        "State": "Stopped"
      },
      {
        "Type": "MainSound",
        "State": "Active"
      },
      {
        "Type": "AlternativeSound",
        "State": "Stopped"
      },
      {
        "Type": "OutputSound",
        "State": "Stopped"
      },
      {
        "Type": "MotionDetection",
        "State": "Stopped"
      },
      {
        "Type": "IO",
        "State": "Stopped"
      },
      {
        "Type": "ArchiveVideo",
        "State": "Stopped"
      },
      {
        "Type": "ArchiveSound",
        "State": "Stopped"
      },
      {
        "Type": "SecondAlternativeVideo",
        "State": "Stopped"
      },
      {
        "Type": "ThirdAlternativeVideo",
        "State": "Stopped"
      },
      {
        "Type": "SecondAlternativeSound",
        "State": "Stopped"
      }
    ]
  }
]

```

```

    },
    {
      "Type": "ThirdAlternativeSound",
      "State": "Stopped"
    },
  ],
]

```

Где:

Параметр	Подпараметр	Описание
Id	–	Уникальный идентификатор канала
IsRecordingOn	–	Состояние записи архива канала. Возможные значения: true – запись в архив разрешена, false – запрещена
StreamStates	Type	Тип канала связи. Возможные значения: <ul style="list-style-type: none"> • MainVideo — приём видео, поток Основной; • AlternativeVideo — приём видео, поток Дополнительный 1; • MainSound — прием звука, поток Основной; • AlternativeSound — прием звука, поток Дополнительный 1; • OutputSound — передача звука; • MotionDetection — встроенный детектор движения; • IO — цифровые входы/выходы; • ArchiveVideo — архивное видео; • ArchiveSound — архивный звук; • SecondAlternativeVideo — приём видео, поток Дополнительный 2; • ThirdAlternativeVideo — приём видео, поток Дополнительный 3; • SecondAlternativeSound — приём звука, поток Дополнительный 2; • ThirdAlternativeSound — приём звука, поток Дополнительный 3.
	State	Состояние канала связи. Возможные значения: <ul style="list-style-type: none"> • Stopped — поток остановлен, т.к. он не требуется системе; • Active — поток находится в состоянии получения кадров и событий; • NoConnection — в потоке произошел обрыв соединения с устройством.
IsAnalogCameraConnected	–	Флаг типа камеры, использующейся каналом. Возможные значения: true – устройство подключено как аналоговая камера, false – как цифровое устройство

HTTP-интерфейс для получения событий

Помимо получения информации о текущей конфигурации системы, HTTP-интерфейс позволяет также получать информацию об обнаруженных и созданных системой событиях.

Такие запросы могут поддерживать два формата структуры: обращение к ресурсу с указанием типа запроса и без него.

Обращение к ресурсу без указания типа запроса:

```
{Протокол}://{Сервер}:{Порт}/{Ресурс}?login={Логин}&password={Пароль}&{Параметр}={Значение параметра}
```

Обращение к ресурсу с указанием типа запроса:

```
{Протокол}://{Сервер}:{Порт}/{Ресурс}?type={Тип}login={Логин}&password={Пароль}&{Параметр}={Значение параметра}
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	8080	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https
Ресурс	–	URI ресурса сервера, к которому выполняется запрос
Тип	–	Тип запроса к используемому ресурсу, если требуется
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос. Пользователь должен обладать правами доступа к каналам, функциям и возможностям системы, к которым обращается в рамках запроса
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
Параметр	–	Дополнительный параметр, уточняющий сам запрос или ответ на него. В зависимости от запроса может быть допустимо применение нескольких дополнительных параметров одновременно
Значение параметра	–	Значение применённого дополнительного параметра

Получение списка всех зарегистрированных в системе событий



Реализовано в Macroscop версии 2.1 и более поздних



Доступно получение ответа в формате JSON.

Для фильтрации информации при выполнении запросов на получение событий можно использовать идентификаторы типов событий. Идентификаторы статичны и неизменны независимо от системы, сервера в неё или версии установленного продукта.

Для получения полного списка существующих в системе типов событий необходимо выполнить запрос к ресурсу **command** с типом **getallregisteredevents**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Пример запроса данных в формате **xml**:

```
http://127.0.0.1:8080/command?type=getallregisteredevents&login=root&password=
```

Пример запроса данных в формате **json**:

```
http://127.0.0.1:8080/command?type=getallregisteredevents&login=root&password=&responsetype=json
```

Пример фрагмента ответа на запрос в формате **XML**:

```
<EventInfo>
  <Id>00000000-0000-0000-0000-000000000033</Id>
  <GuiName>Движение</GuiName>
  <AvailabilityModes>
    <HttpEventAvailabilityMode>RealTime</HttpEventAvailabilityMode>
  </AvailabilityModes>
</EventInfo>
```

Пример фрагмента ответа на запрос в формате **JSON**:

```
{
  "Id": "00000000-0000-0000-0000-000000000033",
  "GuiName": "Движение",
  "AvailabilityModes": [
    "RealTime"
  ]
}
```

Некоторые типы событий доступны для получения только в режиме реального времени, в связи с чем их нельзя получить при запросе за прошедшее время. Определить, какие методы получения доступны для выбранного типа события позволяет параметр **AvailabilityModes**.

Параметр может иметь следующие значения:

- **RealTime** — события этого типа доступны для получения в режиме реального времени.
- **SpecialArchive** — события этого типа доступны для получения из архива.

Пример события, доступного для получения как в реальном времени, так и из архива:

```
<EventInfo>
  <Id>427f1cc3-2c2f-4f50-8865-56ae99c3610d</Id>
  <GuiName>Обнаружено лицо (Модуль распознавания лиц)</GuiName>
  <AvailabilityModes>
    <HttpEventAvailabilityMode>RealTime</HttpEventAvailabilityMode>
    <HttpEventAvailabilityMode>SpecialArchive</HttpEventAvailabilityMode>
  </AvailabilityModes>
</EventInfo>
```

Получение событий в реальном времени



Реализовано в Macroscop версии 2.1 и более поздних



Доступно получение ответа в формате JSON.

События можно получать в режиме реального времени в виде непрерывного («бесконечного») HTTP-соединения.



При чтении полученного ответа необходимо учитывать, что информация передаётся с использованием следующих механизмов:

- **Chunked transfer encoding** для передачи динамически формируемого тела ответа. Ввиду "бесконечности" запроса невозможно предугадать точный размер тела ответа, в связи с чем ответ передаётся с заголовком **"Transfer-encoded: chunked"**.
- **Newline delimited JSON streaming** для разделения передаваемых объектов. Разделение объектов в этом механизме осуществляется переводом следующего объекта на новую строку вместо использования символьных разделителей.

Для получения «бесконечного» потока событий системы используется запрос к ресурсу **event**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Опциональный параметр
filter	–	Уникальный идентификатор события. Может быть получен при выполнении запроса Получение списка всех зарегистрированных событий . Опциональный параметр
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

mode	-	Флаг указания серверу генерировать отладочные сообщения при применении значения demo . Опциональный параметр
------	---	--

Пример запроса данных в формате **json**:

```
http://127.0.0.1:8080/event?login=root&password=&responsetype=json
```

Пример фрагмента ответа на запрос в формате **JSON**:

```
{
  "EventId" : "eb0bb455-b85f-4ac4-851f-f30a11797579",
  "Timestamp" : "19.10.2022 09:58:55",
  "BinaryTimestamp" : "5249703721781162729",
  "ZonedTimestamp" : "19.10.2022 09:58:55.377 +05:00",
  "EventDescription" : "Начало движения",
  "IsAlarmEvent" : "False",
  "ChannelId" : "e0391a80-c921-4ffc-9a69-107fcf28e34e",
  "ChannelName" : "Камера 3",
  "Comment" : "",
  "EventType" : "Info",
  "InitiatorName" : "System"
}
{
  "EventId" : "e4b1f78d-35d6-4092-9fd8-72e66de82e01",
  "Timestamp" : "19.10.2022 09:59:11",
  "BinaryTimestamp" : "5249703721937844932",
  "ZonedTimestamp" : "19.10.2022 09:59:11.045 +05:00",
  "EventDescription" : "Окончание движения",
  "IsAlarmEvent" : "False",
  "ChannelId" : "e0391a80-c921-4ffc-9a69-107fcf28e34e",
  "ChannelName" : "Камера 3",
  "Comment" : "",
  "EventType" : "Info",
  "InitiatorName" : "System"
}
```

Параметры запроса **filter** и **channelid** применяются в случаях, когда необходимо получить события только определённого типа и/или с определённого канала. Оба параметра могут быть использованы в одном запросе одновременно, но каждый параметр при этом может иметь только одно значение.

Пример корректного запроса:

```
http://127.0.0.1:8080/event?login=root&password=&channelid=e0391a80-c921-4ffc-9a69-107fcf28e34e&filter=00000000-0000-0000-0000-000000000033&responsetype=json
```

Пример некорректного запроса:

```
http://127.0.0.1:8080/event?login=root&password=&channelid=e0391a80-c921-4ffc-9a69-107fcf28e34e&filter=00000000-0000-0000-0000-000000000033,e4b1f78d-35d6-4092-9fd8-72e66de82e01&responsetype=json
```

При добавлении к запросу параметра **mode=demo** система начнёт генерировать виртуальные события с типом события «Обнаружен автономер». Такие события не сохраняются в журнал событий системы, не ассоциируются с какой-либо камерой и не несут какой-либо действительной информации. Этот параметр может быть полезен для изучения, тестирования и отладки механизмов получения и чтения событий.

Пример запроса виртуальных событий в формате **JSON**:

```
http://127.0.0.1:8080/event?login=root&password=&mode=demo&responsetype=json
```

Пример **виртуального** события в формате **JSON**:

```
{
  "EventId" : "c9d6d086-c965-4cf8-af6-85b3894e3a4a",
  "Timestamp" : "19.10.2022 10:24:02",
  "BinaryTimestamp" : "5249703736847974593",
  "ZonedTimestamp" : "19.10.2022 10:24:02.058 +05:00",
  "EventDescription" : "Обнаружен автономер",
  "IsAlarmEvent" : "False",
  "ChannelId" : "00000000-0000-0000-0000-000000000000",
  "ChannelName" : "",
  "Comment" : "",
  "EventType" : "Info",
  "InitiatorName" : "System",
  "IsIdentified" : "False",
  "plateText" : "",
  "Speed" : "0",
  "Reliability" : "0",
  "Left" : "0",
  "Top" : "0",
  "lastName" : "",
  "firstName" : "",
  "patronymic" : "",
  "carbrand" : "",
  "carcolor" : "",
  "additionalInfo" : "",
  "groups" : "",
  "direction" : ["Unknow"],
  "ExternalId" : "",
  "ExternalOwnerId" : "",
  "Width" : "0",
  "Height" : "0",
  "Numberplate" : ""
}
```

В случае полного или временного отсутствия событий, соответствующих отправленному запросу, система будет возвращать в ответ **KeepAlive** сообщения для сохранения установленного соединения активным. Сообщение такого типа будет отображаться в потоке даже если в запросе был задан фильтр по типу события.

Пример **KeepAlive** сообщения в формате **JSON**:

```
{
  "EventId" : "e9e7a69c-7ee2-3fee-a530-9f8a88124fcc",
  "Timestamp" : "19.10.2022 09:59:19",
  "BinaryTimestamp" : "5249703722021050198",
  "ZonedTimestamp" : "19.10.2022 09:59:19.366 +05:00",
  "EventDescription" : "",
  "IsAlarmEvent" : "False",
  "ChannelId" : "00000000-0000-0000-0000-000000000000",
  "ChannelName" : "",
  "Comment" : "KeepAlive",
  "EventType" : "Info",
  "InitiatorName" : "System"
}
```

В результате работы некоторых модулей аналитики системой регистрируются координаты обнаруженного на кадре объекта (например, координаты лица для модулей Распознавание лиц). При запросе событий такого типа, в теле ответа передаются относительные координаты рамки объекта в виде позиции на кадре левой верхней точки рамки (**Top, Left**), а также её ширины и высоты (**Width, Height**). Отсчёт координат осуществляется от левого верхнего угла кадра.

Пример ответа с координатами объекта:

```
{
  "InitiatorName" : "ExternalEvent",
  "EventId" : "427f1cc3-2c2f-4f50-8865-56ae99c3610d",
  "EventType" : "Info",
  "IsAlarmEvent" : "False",
  "ChannelId" : "89402bc1-2ca3-4aa2-ac06-6c5d8470f1fa",
  "ChannelName" : "Камера 1",
  "Comment" : "",
  "Timestamp" : "22.08.2023 04:44:05",
  "BinaryTimestamp" : "5249968780880794756",
  "ZonedTimestamp" : "22.08.2023 04:44:05.340 +05:00",
  "EventDescription" : "Обнаружено лицо (Модуль распознавания лиц)",
  "FaceId" : "5821fc2b-c9d2-4d72-aa85-6ffe114b7fec",
  "IsIdentified" : "False",
  "lastName" : "",
  "firstName" : "",
  "patronymic" : "",
  "groups" : "",
  "additionalInfo" : "",
  "Left" : "0,1947916597127915",
  "Top" : "0,062500000000000006",
  "Width" : "0,5624999999999999",
  "Height" : "0,9374999999999999",
  "Similarity" : "0",
  "Age" : "25",
  "Gender" : "Male",
  "ExternalId" : "",
  "TemperatureDegreesCelsius" : "0",
  "ImageBytes" : "",
  "Emotion" : "Neutral",
  "EmotionConfidence" : "0,8842294216156006",
  "IsFaceCovered" : "False",
  "IsRotated" : "False",
  "TrajectoryId" : "da758802-6ee6-401f-b13f-b67d969a07c9"
}
```

Для сортировки по времени возникновения события каждый объект имеет временную метку, представленную в теле ответа одновременно в трёх форматах:

- **Timestamp** - дата и время сервера в формате UTC без учёта часового пояса сервера;
- **ZonedTimestamp** - дата и время сервера в формате UTC с указанием часового пояса сервера;
- **BinaryTimestamp** - дата и время сервера в бинарном представлении (метод `DateTime.ToBinary`).



Бинарное представление временной метки имеет наибольшую точность и удобство интеграции, но усложняет задачу чтения временной метки человеком. Для преобразования бинарного представления даты и времени в формат UTC используйте метод `DateTime.FromBinary`.

Получение событий из архива



Реализовано в Macroscop версии 4.0 и более поздних



Доступно получение ответа в формате JSON.

Возможность получения событий из архива позволяет реализовать в сторонних приложениях собственный Журнал событий на основе полученных данных.



Ответ на запрос будет содержать все типы событий, доступные в Журнале, а также детали события модуля распознавания лиц — Распознано лицо.



Запрос состоит из URL и тела запроса, передаваемого в формате JSON. Для успешного выполнения необходимо, чтобы приложение-отправитель умело работать с такими запросами.

Для получения списка событий из архива используется **GET**-запрос к ресурсу **archive_events**.

Доступны дополнительные параметры URL:

Параметр	Значение по умолчанию	Описание
shortevent	false	Флаг, указывающий серверу на необходимость отправки укороченного варианта сообщения о событии
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Базовый запрос позволяет выполнить обращение к URL без заполнения тела запроса.

Пример URL:

```
http://127.0.0.1:8080/archive_events?login=root&password=
```

В ответ на данный запрос будет передан полный список событий системы без применения фильтрации.

Для уточнения запроса необходимо добавить к обращению дополнительные параметры, указав их в теле запроса в формате **JSON**.



Если параметр не указан в теле запроса, для него будет применено значение по умолчанию при выполнении запроса.

Дополнительные параметры тела запроса:

Параметр	Тип	Значение по умолчанию	Описание
StartTimeUtc	string	Минимально возможное время	Время начала поиска (UTC) в формате dd.MM.yyyy HH:mm:ss.fff
EndTimeUtc	string	Максимально возможное время	Время конца поиска (UTC) в формате dd.MM.yyyy HH:mm:ss.fff

EventCategories	number	Все категории	Список категорий события: 0 - информация 1 - тревога 2 - ошибка
EventInitiatorTypes	number	Все типы инициаторов событий	Список типов инициаторов событий: 0 - система 1 - пользователь 2 - сценарий 3 - пользовательский сценарий 4 - задача по расписанию 8 - внешний модуль
EventInitiators	string	Все пользователи	Список идентификаторов (GUID) пользователей системы
EventIds	string	Все события	Список идентификаторов (GUID) событий
ChannelIds	string	Все камеры и пустой Guid	Список идентификаторов (GUID) камер. Для получения событий, не привязанных к камере, необходимо отправить пустой GUID (00000000-0000-0000-0000-000000000000)
IsSearchFromBegin	boolean	false	Поиск событий в архиве с начала, а не с конца временного промежутка
SearchLimitCount	number	5000	Ограничение числа возвращаемых записей



При отправке некорректных значений для следующих параметров будет получен ответ с кодом ошибки **400 (Bad Request)**:

- StartTimeUtc
- EndTimeUtc
- IsSearchFromBegin
- SearchLimitCount

Пример тела запроса с дополнительными параметрами в формате **JSON**:

```
{
  "startTimeUtc": "17.04.2023 09:54:31.775",
  "endTimeUtc": "17.04.2023 10:54:31.775",
  "cameraIds": ["00000000-0000-0000-0000-000000000000"],
  "eventCategories": [0,1,2],
  "eventInitiatorTypes": [0,2,8,4,1,3],
  "eventInitiators": ["2aa0118f-8849-499c-b0df-6b071d95ee66"],
  "isSearchFromBegin": false,
  "searchLimitCount": 200
}
```

В ответ сервер вернёт информацию о событии одного из представленных ниже форматов, в зависимости от дополнительных параметров запроса и свойств самого события.

Пример события без дополнительной информации:

```
{
  "ChannelId": "00000000-0000-0000-0000-000000000000",
  "ChannelName": "",

```

```

"Event": null,
"EventCategory": 0,
"EventComment": "Скорость записи на диск: 1,59 МБайт/с.Подсистема работы с архивом.",
"EventDescription": null,
"EventId": "00000000-0000-0000-0000-000000000010",
"EventInitiatorType": 0,
"Timestamp": "2022-10-03T08:45:20.9497012Z"
}

```

Пример события с дополнительной информацией:

```

{
  "ChannelId": "effbcd69-9f89-4301-87bf-2663bfff0a44d",
  "ChannelName": "Камера 9",
  "Event": {
    "AdditionalInfo": "Лена. Менеджмент",
    "Age": 24,
    "Emotion": 2,
    "EventName": "FaceDetectedNotifyEvent",
    "EventTime": "2022-10-03T08:23:40.8019552Z",
    "FaceId": "fe705eb2-4f76-47eb-92eb-178c0ccf7077",
    "FaceImageBase64": "base64 jpeg image",
    "FirstName": "Лена",
    "Gender": 1,
    "Groups": ["Доверенная", "Перехват"],
    "IsIdentified": true,
    "LastName": "Иванова",
    "Patronymic": "Сергеевна",
    "Height": 0.152734375,
    "Left": 0.56064453125,
    "Top": 0.3083984375,
    "Width": 0.11455078125,
    "Similarity": 0.9900000095367432,
  },
  "EventCategory": 1,
  "EventComment": "Иванова Лена Сергеевна",
  "EventDescription": "Обнаружено лицо (Модуль распознавания лиц)",
  "EventId": "427f1cc3-2c2f-4f50-8865-56ae99c3610d",
  "EventInitiatorType": 0,
  "Timestamp": "2022-10-03T08:23:40.8019552Z"
}

```

При указании в URL запроса параметра **shorthevent=true** сервер вернёт укороченный текст события.

Пример запроса укороченного текста события:

```

http://127.0.0.1:8080/archive\_events?login=root&password=&shorthevent=true

```

Пример ответа с укороченным текстом события:

```

{
  "EventCategory": 1,
  "EventId": "427f1cc3-2c2f-4f50-8865-56ae99c3610d",
  "Timestamp": "2022-10-03T08:23:40.8019552Z"
}

```

В тексте события, передаваемом в теле возвращаемого сервером ответа, могут встречаться следующие параметры:

Параметр	Тип	Описание
ChannelId	Guid	Идентификатор камеры
ChannelName	string	Название камеры
EventId	Guid	Идентификатор типа события
EventCategory	number	Категория события: 0 - информация 1 - тревога 2 - ошибка
EventComment	string	Комментарий события
EventDescription	string	Описание события
EventInitiatorType	number	Тип инициатора события 0 - система 1 - пользователь 2 - сценарий 3 - пользовательский сценарий 4 - задача по расписанию 8 - внешний модуль
Timestamp	string	Время события в UTC
Event	Для Unified событий – null, для аналитических модулей – дополнительная информация о событии (если существует)	Вложенный объект с информацией о событии

Для получения списка зарегистрированных в системе типов событий, которые могут быть выданы в ответ на запрос к ресурсу **archive_events**, используется запрос к ресурсу **archive_event_types**.



Запрос к этому ресурсу не предполагает отправки дополнительного тела запроса в формате JSON.

Доступны дополнительные параметры URL:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/archive_event_types?login=root&password=&responsetype=json
```

В ответ на данный запрос сервер вернёт список событий в формате JSON следующего вида:


```
[
  {
    "Id": "7da43299-9c0d-4043-9183-e947b5f6bdca",
    "Name": "Громкий звук"
  }
]
```

Получение списка специальных архивных событий



Реализовано в Macroscop версии 2.1 и более поздних



Доступно получение ответа в формате JSON.

Ещё одним способом получения событий из архива является порционная выгрузка данных в виде списка событий, зафиксированных системой за заданное время. Для этого применяется запрос к ресурсу **specialarchiveevents**.



При чтении полученного ответа необходимо учитывать, что информация передаётся с использованием механизма **Newline delimited JSON streaming**. Разделение объектов в этом механизме осуществляется переводом следующего объекта на новую строку вместо использования символьных разделителей.



Для данного типа запроса действует ограничение на максимальное количество событий, возвращаемых сервером единоразово – **1000** записей. Для получения последующих событий необходимо начинать поиск от времени последнего полученного события.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
starttime	–	Дата и время начала интервала поиска событий. Указывается по часовому поясу UTC в формате DD-MM-YYYY+hh:mm:ss. Обязательный параметр
endtime	–	Дата и время конца интервала поиска событий. Указывается по часовому поясу UTC в формате DD-MM-YYYY+hh:mm:ss. Обязательный параметр
eventid	–	Уникальный идентификатор типа события. Может быть получен при выполнении запроса Получение списка зарегистрированных событий . Обязательный параметр
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Опциональный параметр
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml, json

Пример запроса данных в формате **JSON**:

```
http://127.0.0.1:8080/specialarchiveevents?login=root&password=&starttime=19.10.2022+10:50:00&endtime=19.10.2022+11:00:00&eventid=b0536c2f-2f09-4969-bf1a-9fb847b87d21&responsetype=json
```

Пример фрагмента ответа на запрос в формате **JSON**:

```
{
  "EventId" : "b0536c2f-2f09-4969-bf1a-9fb847b87d21",
  "Timestamp" : "19.10.2022 10:59:56",
  "BinaryTimestamp" : "5249703758396001539",
  "ZonedTimestamp" : "19.10.2022 10:59:56.861 +05:00",
  "EventDescription" : "Событие трекинга",
  "IsAlarmEvent" : "True",
  "ChannelId" : "e0391a80-c921-4ffc-9a69-107fcf28e34e",
  "ChannelName" : "Камера 3",
  "Comment" : "Движение в зоне",
  "EventType" : "Alarm",
  "InitiatorName" : "ExternalEvent",
  "alertType" : ["MovingInZone"],
  "AlertTime" : "638017739968613635",
  "TrajectoryId" : "57f88149-9976-4953-9b4c-3921c689b82d",
  "Left" : "0,0032153846710991974",
  "Top" : "0,5185692308091415",
  "Width" : "0,05787692314846298",
  "Height" : "0,13713846175798144"
}
```

Получение из архива распознанных автомобильных номеров

{...} Доступно получение ответа в формате JSON.

Для модуля Распознавания автомобильных номеров существует дополнительный тип запроса, позволяющий получить список распознанных за указанный период времени номеров в более коротком формате, чем с помощью ресурсов [event](#) или [specialarchivevents](#).

Для получения списка распознанных автомобильных номеров в укороченном виде используется запрос к ресурсу **autovprs_export**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
starttime	–	Время начала интервала поиска событий. Указывается по часовому поясу сервера в формате YYYY-MM-DD-hh-mm-ss-fff. Обязательный параметр
finishtime	–	Время конца интервала поиска событий. Указывается по часовому поясу сервера в формате YYYY-MM-DD-hh-mm-ss-fff. Обязательный параметр
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр



Ответ от сервера для данного запроса всегда будет в формате JSON.

Пример запроса:

```
http://127.0.0.1:8080/autovprs_export?login=root&password=&channelid=f2e18694-3f58-4128-a48e-37dd184b109b&startTime=2022-10-20-16-40-00-000&finishTime=2022-10-20-17-00-00-000
```

Пример ответа:

```
[
  {
    "TimeUtc" : "20.10.2022 11:48:11.707",
    "Numberplate" : "X532MH159",
    "LastName" : "",
    "FirstName" : "",
    "PatronymicName" : "",
    "Group" : "",
    "Direction" : "Unknown"
  },
  {
    "TimeUtc" : "20.10.2022 11:48:35.097",
    "Numberplate" : "C112HP72",
    "LastName" : "",
    "FirstName" : "",
    "PatronymicName" : "",
    "Group" : "",
    "Direction" : "Unknown"
  }
]
```



При отправке запроса и чтении ответа необходимо учитывать особенность отображения времени.

В теле запроса временной интервал указывается по часовому поясу сервера в формате ГГГГ-ММ-ДД-чч-мм-сс-мс.

В теле ответа время распознавания автомобильного номера указывается по часовому поясу UTC(+0) в формате ДД-ММ-ГГГГ чч-мм-сс.

Таким образом, если запрос отправляется на сервер с часовым поясом UTC+3 с целью получить список распознанных номеров за интервал 15:00:00 - 15:10:00, время в теле запроса указывается согласно требуемому интервалу, тогда как в теле ответа соответствующие запросу номера будут иметь время из интервала 12:00:00 - 12:10:00.

HTTP-интерфейс для выполнения команд сервером Macroscop

HTTP-интерфейс **Macroscop** позволяет не только получать сведения о системе, но и управлять некоторыми её элементами, выполняя описанные ниже CGI-запросы.

Такие запросы, за одним исключением, имеют следующий формат:

```
{Протокол}://{Сервер}:{Порт}/command?type={Тип}&login={Логин}&password={Пароль}&{Параметр}={Значение параметра}
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	8080	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https
Тип	–	Тип запроса к используемому ресурсу, если требуется
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос. Пользователь должен обладать правами доступа к каналам, функциям и возможностям системы, к которым обращается в рамках запроса
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
Параметр	–	Дополнительный параметр, уточняющий сам запрос или ответ на него. В зависимости от запроса может быть допустимо применение нескольких дополнительных параметров одновременно
Значение параметра	–	Значение применённого дополнительного параметра

Для большинства запросов на выполнение действий сервер передаёт однотипный ответ о статусе выполнения запроса. Если запрос был успешно выполнен, тело ответа будет содержать только статус **OK** или **Success**. В противном случае в теле ответа будет дана информация о возникшей ошибке выполнения.

Включение/выключение записи на канале

Для ручного включения записи архива на канале используется тип запроса **recording**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр

mode	–	Режим записи архива. Возможные значения: start — начать запись, stop — остановить запись. Обязательный параметр
interval	–	Время записи в минутах. Обязательный параметр при mode=start

Пример запроса для включения записи продолжительностью 15 минут:

```
http://127.0.0.1:8080/command?type=recording&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&login=root&password=&mode=start&interval=15
```

Пример запроса для выключения записи:

```
http://127.0.0.1:8080/command?type=recording&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&login=root&password=&mode=stop
```



Данный запрос не оказывает влияние на каналы, для которых запись архива отключена или уже ведётся постоянно.

Изменение времени сервера Macroscop

Для принудительной установки времени на устройстве, где установлен сервер **Macroscop**, используется тип запроса **settime**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
time	–	Дата и время в часовом поясе UTC. Обязательный параметр



Параметр **time** поддерживает несколько форматов записи даты и времени:

- Дата может записываться как с полным обозначением года («YYYY или 2023»), так и с сокращённым («YY» или «23»).
- Для разделения числа, месяца и года в дате можно использовать как точку («.»), так и дефис («-»).
- Допускается написание даты, начиная как с числа («DD.ММ.YYYY» или «14.06.2023»), так и с года («YYYY.ММ.DD» или «2023.06.14»).
- Разделение даты и времени может осуществляться как пробелом (« ») или его кодом («%20»), так и плюсом («+»).

Пример запроса:

```
http://127.0.0.1:8080/command?type=settime&login=root&password=&time=14.06.2023+08:11:00
```

Установка профиля экрана на клиенте


Для передачи на подключенный к серверу **Macroscop** Клиент команды о смене профиля экрана используется тип запроса **setprofile**.



С помощью данного запроса можно устанавливать только серверные профили экрана. Профили (виды), созданные непосредственно на клиенте, запросом не поддерживаются.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0. Обязательный параметр
profileid	–	Идентификатор профиля экрана для установки. Список доступных профилей может быть получен с помощью запроса Получение списка серверных профилей экрана



-  Значение **clientip** должно содержать фактический адрес клиента **Macroscop**. Использование localhost (127.0.0.1) вызовет ошибку даже при обращении к клиенту, установленному непосредственно на сервере.

Пример запроса:

```
http://127.0.0.1:8080/command?type=setprofile&login=root&password=&clientip=192.168.1.2&monitor=0&profile=13851f3d-c7d3-4ec6-b0ff-2d66873bf118
```


Смена сетки на клиенте

Для изменения сетки камер в рамках действующего профиля экрана на выбранном клиенте **Macroscop** используется тип запроса **setgrid**.

-  Запрос позволяет изменять сетки только для тех профилей экранов, которые были созданы непосредственно на клиентском устройстве. Изменение серверных профилей не поддерживается.
-  Любые изменения сетки сохраняются автоматически в рамках действующего профиля экрана. Новый профиль при этом не создаётся.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0. Обязательный параметр
cells	–	Имя устанавливаемой сетки. Имена доступных сеток можно получить с помощью запроса Получение списка доступных сеток



-  Значение **clientip** должно содержать фактический адрес клиента **Macroscop**. Использование localhost (127.0.0.1) вызовет ошибку даже при обращении к клиенту, установленному непосредственно на сервере.

Пример запроса:

```
http://127.0.0.1:8080/command?type=setgrid&login=root&password=&clientip=192.168.1.2&monitor=0&cells=25
```



Установка канала в ячейку сетки

Для размещения нового канала в ячейке сетки камер действующего профиля экрана используется тип запроса **setcell**.

-  Запрос позволяет изменять сетки только для тех профилей экранов, которые были созданы непосредственно на клиентском устройстве. Изменение серверных профилей не поддерживается.
-  Любые изменения сетки сохраняются автоматически в рамках действующего профиля экрана. Новый профиль при этом не создаётся.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0. Обязательный параметр
cell	–	Номер ячейки. Отсчёт начинается с 0. Обязательный параметр
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
mode	realtime	Режим воспроизведения канала в ячейке. Возможные значения: realtime – воспроизведение потока реального времени, archive – воспроизведение архива
starttime	Момент получения запроса	Начальная точка воспроизведения архива. Применяется только при mode=archive
speed	1	Скорость воспроизведения архива. Применяется только при mode=archive . Возможные значения: 0,1; 0,2; 0,5; 1; 2; 5; 10; 20; 60; 120



-  Значение **clientip** должно содержать фактический адрес клиента **Macroscop**. Использование localhost (127.0.0.1) вызовет ошибку даже при обращении к клиенту, установленному непосредственно на сервере.
-  Параметр **searchTime** поддерживает несколько форматов записи даты и времени:
 - Дата может записываться как с полным обозначением года («YYYY или 2023»), так и с сокращённым («YY» или «23»).
 - Для разделения числа, месяца и года в дате можно использовать как точку («.»), так и дефис («-»).
 - Допускается написание даты, начиная как с числа («DD.ММ.YYYY» или «14.06.2023»), так и с года («YYYY.ММ.DD» или «2023.06.14»).
 - Разделение даты и времени может осуществляться как пробелом (« ») или его кодом («%20»), так и плюсом («+»).

Пример запроса:

```
http://127.0.0.1:8080/command?type=setcell&login=root&password=&clientip=192.168.1.2&monitor=0&cell=4&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&mode=archive&starttime=14.09.2023+10:10:00&speed=2
```


Удаление канала из ячейки сетки

Для удаления канала из одной ячейки сетки камер в рамках действующего профиля экрана используется тип запроса **clearcell**.

-  Запрос позволяет изменять сетки только для тех профилей экранов, которые были созданы непосредственно на клиентском устройстве. Изменение серверных профилей не поддерживается.
-  Любые изменения сетки сохраняются автоматически в рамках действующего профиля экрана. Новый профиль при этом не создаётся.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0. Обязательный параметр
cell	–	Номер ячейки. Отсчёт начинается с 0. Обязательный параметр



-  Значение **clientip** должно содержать фактический адрес клиента **Macroscop**. Использование localhost (127.0.0.1) вызовет ошибку даже при обращении к клиенту, установленному непосредственно на сервере.

Пример запроса:

```
http://127.0.0.1:8080/command?type=cleargrid&login=root&password=&clientip=192.168.1.2&monitor=0&cell=4
```

Очистка сетки камер

Для удаления всех используемых каналов из ячеек сетки камер в рамках действующего профиля экрана используется тип запроса **cleargrid**.

-  Запрос позволяет изменять сетки только для тех профилей экранов, которые были созданы непосредственно на клиентском устройстве. Изменение серверных профилей не поддерживается.
-  Любые изменения сетки сохраняются автоматически в рамках действующего профиля экрана. Новый профиль при этом не создаётся.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0. Обязательный параметр



Значение **clientip** должно содержать фактический адрес клиента **Macroscop**. Использование localhost (127.0.0.1) вызовет ошибку даже при обращении к клиенту, установленному непосредственно на сервере.

Пример запроса:

```
http://127.0.0.1:8080/command?type=cleargrid&login=root&password=&clientip=192.168.1.2&monitor=0
```

Постановка канала на охрану



Реализовано в Macroscop версии 2.1 и более поздних.

Для изменения состояния режима охраны для канала используется тип запроса **setguardian**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0. Обязательный параметр
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
isguardmodeenabled	–	Переключение состояния режима охраны. Возможные значения: true – включить режим охраны, false – выключить. Обязательный параметр

Пример запроса:

```
http://127.0.0.1:8080/command?type=setguardian&login=root&password=&clientip=192.168.1.2&monitor=0&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&isguardianmodeenabled=true
```

Отправка звука на камеру



Реализовано в Macroscop версии 2.1 и более поздних.

Macroscop поддерживает возможность отправки звуковых потоков для воспроизведения через динамик камеры. Для этого применяется POST-запрос к ресурсу **sendsound**. В URL запроса указываются параметры цели, в теле запроса передаются аудиоданные.



Данный запрос предназначен в первую очередь для использования в сторонних приложениях, выполняющих роль **клиента**. Для выполнения данного запроса может потребоваться использование дополнительных библиотек.



Запрос состоит из URL и тела запроса, передаваемого в формате JSON. Для успешного выполнения необходимо, чтобы приложение-отправитель умело работать с такими запросами.

Дополнительные параметры URL:

Параметр	Значение по умолчанию	Описание
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
clientid	–	Уникальный идентификатор сеанса передачи



В качестве значения параметра **clientid** должен использоваться **GUID**, формируемый приложением, выполняющим роль **клиента**.

Пример URL:

```
http://127.0.0.1:8080/sendsound?login=root&password=&channelid=66abc0c4-d4b7-4d71-8ed1-e7beadf0dc46&clientid=66abc0c4-d4b7-4d71-8ed1-e7beadf0dc46
```

Помимо перечисленных выше параметров URL, выполняемый POST-запрос также должен содержать заголовок **ContentType = "multipart/form-data;"** для указания типа данных, передаваемых в теле запроса.

Дополнительная информация:

- Передаваемые данные должны отвечать определённым требованиям. Для формирования порции аудиоданных (кодирования звука) рекомендуется использовать сторонние библиотеки (например, NAudio: <https://github.com/naudio/NAudio>), задав при кодировании следующие параметры: Samplesrate=8000; Bitspersample=16; Количество каналов = 1.
- Запрос не является постоянным. Передача данных начинается только после того, как сервером от приложения-клиента будет получен соответствующий запрос.
- Запрос не является непрерывным. В рамках одного запроса от сервера камере передаётся одна порция аудиоданных, после чего запрос считается исполненным.
- Сеанс передачи может состоять из нескольких запросов. Сеансом передачи считается серия запросов с определённого клиентского устройства к определённой камере. В рамках одного сеанса передачи допускается использовать один и тот же **clientid** для снижения серверных издержек.
- Каждый сеанс передачи должен иметь уникальный **clientid**. Генерация **GUID** для **clientid** должна выполняться приложением, выполняющим роль **клиента**.

Генерация события из внешней системы



Реализовано в Macroscop версии 2.1 и более поздних.

Для создания в журнале системы события с типом Событие из внешней системы используется тип запроса **generateexternalevent**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
systemname	–	Название внешней системы
information	–	Строка с информацией о событии

eventcode	–	Числовой код события
-----------	---	----------------------




Данный запрос предназначен преимущественно для реализации базовой интеграции с любой сторонней системой, не поддерживаемой **Macroscop**, включая собственные разработки. Значения параметров **systemname**, **information** и **eventcode** могут быть любыми.



Пример запроса:

```
http://127.0.0.1:8080/command?type=generateexternalevent&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&systemname=TestSystem&information=record&eventcode=5
```

В **Журнале событий** приложения **Macroscop Клиент** данное событие будет выглядеть следующим образом:

События		
Время	Тип	Описание события
21.10.2022 15:07:53		Камера 15. Внешнее событие от системы testsystem. Код события 5. Информация: alarm.
		<p>Время: 21 октября 2022, 15:07:53.137 Камера: Камера 15. Тип: Тревога. Событие: Событие из внешней системы. Инициатор: Система. Описание: Камера 15. Внешнее событие от системы testsystem. Код события 5. Информация: alarm.</p>

Также на эти события можно назначать действия в сценариях (посредством приложения **Macroscop Конфигуратор**):

 **Добавление задачи по событию в системе**


Событие в системе

Событие из внешней системы
▼

Действие




Генерация тревоги
▼

Применить

Отмена

Настройка действия ✕

Действия Условия выполнения (3)

Атрибут события	Сравнение	Значение	
Название системы	Равно	TestSystem	
Информация	Равно	Alarm	
Код события	Равно	5	

[Добавить условие](#)

Применить **Отмена**

HTTP-интерфейс для управления PTZ-возможностями

В дополнение к возможности управления компонентами **Macroscop**, HTTP-интерфейс также позволяет получать информацию и передавать команды управления подключенным к системе PTZ-устройствам.

Такие запросы в общем виде имеют следующий формат:

```
{Протокол}://{Сервер}:{Порт}/ptz?command={Тип}&login={Логин}&password={Пароль}&channelid={Канал}&{Параметр}={Значение параметра}
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	8080	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https
Тип	–	Тип запроса к используемому ресурсу, если требуется
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос. Пользователь должен обладать правами доступа к каналам, функциям и возможностям системы, к которым обращается в рамках запроса
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
Канал	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы
Параметр	–	Дополнительный параметр, уточняющий сам запрос или ответ на него. В зависимости от запроса может быть допустимо применение нескольких дополнительных параметров одновременно
Значение параметра	–	Значение применённого дополнительного параметра



В отличие от команд управления компонентами **Macroscop**, перечисленные ниже команды передаются через сервер на PTZ-устройство. Возможность и точность их выполнения зависит непосредственно от устройства и его интеграции в **Macroscop**.

Для запросов, содержащих в себе команды управления PTZ-возможностями устройства, сервер возвращает однотипные ответы о результатах попытки передачи запроса на устройство. Если запрос был успешно передан, тело ответа будет содержать только статус **OK**. В противном случае в теле ответа будет дана информация о возникшей ошибке передачи.

Получение информации о возможностях устройства

{...} Доступно получение ответа в формате JSON.

Для получения информации о поддерживаемых системой опциях управления PTZ функционалом устройства используется тип запроса **getcapabilities**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml, json

Пример запроса с ответом в формате **XML**:

```
http://127.0.0.1:8080/ptz?command=getcapabilities&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b
```

Пример запроса с ответом в формате **JSON**:

```
http://127.0.0.1:8080/ptz?command=getcapabilities&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&login=root&password=&responsetype=json
```

В ответ сервер передаст список PTZ-возможностей, поддерживаемых системой, с указанием через значения параметров тех возможностей, которые поддерживаются самим устройством.

Пример ответа на запрос в формате **XML**:

```
<CameraCapabilities>
  <PtzCapabilities>
    <HomePositionSupports>true</HomePositionSupports>
    <MoveToSupports>true</MoveToSupports>
    <AreaZoomSupports>true</AreaZoomSupports>
    <ZoomSupports>true</ZoomSupports>
    <ContiniousZoomSupports>true</ContiniousZoomSupports>
    <AbsoluteZoomSupports>>false</AbsoluteZoomSupports>
    <MaxMinZoomSupports>>false</MaxMinZoomSupports>
    <AutoFocusSupports>true</AutoFocusSupports>
    <ManualFocusSupports>true</ManualFocusSupports>
    <ContiniousFocusSupports>true</ContiniousFocusSupports>
    <AutoIrisSupports>>false</AutoIrisSupports>
    <ManualIrisSupports>>false</ManualIrisSupports>
    <InfraredLightSupports>true</InfraredLightSupports>
    <WipersControlSupports>true</WipersControlSupports>
    <WasherControlSupports>true</WasherControlSupports>
    <SupportedStepMoveDirections>None</SupportedStepMoveDirections>
    <SupportedContiniousMoveDirections>Any</SupportedContiniousMoveDirections>
    <SetRelativePositionSupports>>false</SetRelativePositionSupports>
    <SetAbsolutePositionSupports>>false</SetAbsolutePositionSupports>
    <GetAbsolutePositionSupports>>false</GetAbsolutePositionSupports>
```

```

    <BlackWhiteModeSupports>false</BlackWhiteModeSupports>
    <NightModeSupports>false</NightModeSupports>
  </PtzCapabilities>
  <Resolution>
    <Width>1920</Width>
    <Height>1080</Height>
  </Resolution>
</CameraCapabilities>

```

Пример ответа на запрос в формате **JSON**:

```

{
  "PtzCapabilities": {
    "HomePositionSupports": true,
    "MoveToSupports": true,
    "AreaZoomSupports": true,
    "ZoomSupports": true,
    "ContiniousZoomSupports": true,
    "AbsoluteZoomSupports": false,
    "MaxMinZoomSupports": false,
    "AutoFocusSupports": true,
    "ManualFocusSupports": true,
    "ContiniousFocusSupports": true,
    "AutoIrisSupports": false,
    "ManualIrisSupports": false,
    "InfraredLightSupports": true,
    "WipersControlSupports": true,
    "WasherControlSupports": true,
    "SupportedStepMoveDirections": "0",
    "SupportedContiniousMoveDirections": "255",
    "SetRelativePositionSupports": false,
    "SetAbsolutePositionSupports": false,
    "GetAbsolutePositionSupports": false,
    "BlackWhiteModeSupports": false,
    "NightModeSupports": false
  },
  "Resolution": {
    "Width": 1920,
    "Height": 1080
  }
}

```

Где:

Параметр	Описание
HomePositionSupports	Поддержка возврата к домашнему положению камеры
MoveToSupports	Поддержка центрирования (перемещения камеры) на заданной точке
AreaZoomSupports	Поддержка функции приближения выделенной области кадра
ZoomSupports	Поддержка «шагового» (однократного) изменения уровня зума
ContiniousZoomSupports	Поддержка «непрерывного» (выполняемого до принудительного завершения) изменения уровня зума
AbsoluteZoomSupports	Поддержка абсолютного зума

MaxMinZoomSupports	Поддержка перехода к максимальному или минимальному уровню зума
AutoFocusSupports	Поддержка функции автофокусировки
ManualFocusSupports	Поддержка ручного управления фокусом
ContiniousFocusSupports	Поддержка «слеящего» фокуса
ManualIrisSupports	Поддержка ручного управления диафрагмой
InfraredLightSupports	Поддержка управления инфракрасной подсветкой
WipersControlSupports	Поддержка управления механическим очистителем
WasherControlSupports	Поддержка управления омывателем
SupportedStepMoveDirections	Маска, описывающая доступные направления для «шагового» перемещения
SupportedContiniousMoveDirections	Маска, описывающая доступные направления для «непрерывного» перемещения
SetRelativePositionSupports	Поддержка возможности указать позицию относительно текущего положения устройства
SetAbsolutePositionSupports	Поддержка возможности указать абсолютную позицию устройства
GetAbsolutePositionSupports	Поддержка возможности получить информацию о текущей абсолютной позиции камеры
BlackWhiteModeSupports	Поддержка переключения устройства в чёрно-белый режим
NightModeSupports	Поддержка переключения устройства в «ночной» режим
Resolution	Текущее разрешение кадра основного потока, где: <ul style="list-style-type: none"> • Width – ширина кадра. • Height – высота кадра.



Для большинства PTZ-возможностей есть лишь два возможных значения: **true**, означающее, что возможность поддерживается, и **false**, означающее обратное. Исключение составляют параметры **SupportedStepMoveDirections** и **SupportedContiniousMoveDirections**, чьё значение принимает числовое выражение.

Получение пресетов устройства



Доступно получение ответа в формате JSON.

Для получения списка предустановленных положений на камере используется тип запроса **getpresets**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml , json

Пример запроса с ответом в формате **XML**:

```
http://127.0.0.1:8080/ptz?command=getpresets&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b
```

Пример запроса с ответом в формате **JSON**:

```
http://127.0.0.1:8080/ptz?command=getpresets&login=root&password=&responsetype=json&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b
```

В ответ сервер вернёт список всех пресетов в том виде, как они созданы на устройстве.

Пример ответа на запрос в формате **XML**:

```
<ArrayOfString>
  <string>Дверь</string>
  <string>Двор</string>
  <string>Окно 1 этаж</string>
  <string>Окно 2 этаж</string>
  <string>Ворота</string>
</ArrayOfString>
```

Пример ответа на запрос в формате **JSON**:

```
[
  "Дверь",
  "Двор",
  "Окно 1 этаж",
  "Окно 2 этаж",
  "Ворота"
]
```



Macroscop получает список пресетов в том виде, как он хранится на самом устройстве. Изменения списка в связи с добавлением, изменением или удалением пресетов зависят от самого устройства. В некоторых случаях добавление нового пресета может изменить порядковые номера уже существующих пресетов.

Установка пресета

Для перевода устройства в одно из предустановленных положений используется тип запроса **gotopreset**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
index	–	Порядковый номер пресета в списке. Отсчёт начинается с 1. Обязательный параметр

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=gotopreset&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&index=1
```

«Непрерывное» движение

Для передачи на устройство команды для непрерывного перемещения в указанном направлении используется тип запроса **startmove**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
panspeed	–	Скорость по горизонтали от -100 до 100. Обязательный параметр
tiltspeed	–	Скорость по вертикали от -100 до 100. Обязательный параметр
stoptimeout	500	Время в миллисекундах, через которое команда будет остановлена



Параметры **panspeed** и **tiltspeed** должны быть указаны в запросе даже при условии, когда движение по одной из осей не требуется. В таком случае незадействованный параметр должен принимать значение **0**.

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=startmove&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&panspeed=2&tiltspeed=2&stoptimeout=100
```

«Непрерывное» изменение фокуса

Для передачи на устройство команды для непрерывного изменения фокуса используется тип запроса **startchangefocus**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
----------	-----------------------	----------

speed	–	Скорость фокусировки от -100 до 100. Обязательный параметр
stoptimeout	500	Время в миллисекундах, через которое команда будет остановлена

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=startchangeofocus&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&speed=5&stoptimeout=100
```

«Непрерывный» зум

Для передачи на устройство команды для непрерывного изменения степени увеличения изображения используется тип запроса **startzoom**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
speed	–	Скорость изменения зума от -100 до 100. Обязательный параметр
stoptimeout	500	Время в миллисекундах, через которое команда будет остановлена

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=startzoom&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&speed=2&stoptimeout=100
```

Остановка «непрерывных» команд

Для прерывания действий камеры, вызванных «непрерывными» командами, используется тип запроса **stop**.

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=stop&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b
```

Автофокусировка

Для включения механизма автоматической фокусировки используется тип запроса **setautofocus**.

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=setautofocus&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b
```

Центрирование

Для центрирования камеры на определённой точке кадра используется тип запроса **moveto**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
width	–	Ширина кадра в пикселях. Обязательный параметр
height	–	Высота кадра в пикселях. Обязательный параметр
x	–	Положение точки центрирования по горизонтали. Может принимать значение от 0 до значения параметра width, отсчёт осуществляется слева направо. Обязательный параметр
y	–	Положение точки центрирования по вертикали. Может принимать значение от 0 до значения параметра height, отсчёт осуществляется сверху вниз. Обязательный параметр

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=moveto&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&width=1920&height=1080&x=300&y=650
```



Для получения информации о ширине и высоте кадра можно использовать запрос на [получение информации о возможностях устройства](#).

«Шаговое» движение

Для передачи на устройство команды для выполнения шагового перемещения в указанном направлении относительно текущего положения используется тип запроса **move**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
tiltstep	–	Шаг по вертикали от -100 до 100. Обязательный параметр
panstep	–	Шаг по горизонтали от -100 до 100. Обязательный параметр



Параметры **tiltstep** и **panstep** должны быть указаны в запросе даже при условии, когда движение по одной из осей не требуется. В таком случае незадействованный параметр должен принимать значение **0**.

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=move&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&tiltstep=15&panstep=0
```

«Шаговый» зум

Для передачи на устройство команды для выполнения шагового изменения степени увеличения изображения используется тип запроса **zoom**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
zoomstep	–	Шаг от -100 до 100. Обязательный параметр

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=zoom&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&zoomstep=10
```

Приближение выделенной области (AreaZoom)

Для приближения определённой части кадра используется тип запроса **showrect**.



Координаты зоны приближения указываются с помощью четырёхугольника, определяемого передаваемыми в запросе параметрами: координаты левого верхнего угла четырёхугольника, а также его ширины и высоты относительно указанного угла.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
frameWidth	–	Ширина кадра в пикселях. Обязательный параметр
frameHeight	–	Высота кадра в пикселях. Обязательный параметр
x	–	Положение левого верхнего угла четырёхугольника по горизонтали. Может принимать значение от 0 до значения параметра frameWidth , отсчёт осуществляется слева направо. Обязательный параметр
y	–	Положение левого верхнего угла четырёхугольника по вертикали. Может принимать значение от 0 до значения параметра frameHeight , отсчёт осуществляется слева направо. Обязательный параметр
width	–	Ширина зоны увеличения в пикселях от значения параметра x . Обязательный параметр
height	–	Высота зоны увеличения в пикселях от значения параметра y . Обязательный параметр



Для получения информации о ширине и высоте кадра можно использовать запрос на [получение информации о возможностях устройства](#).

Пример запроса:

```
http://127.0.0.1:8080/ptz?command=showrect&login=root&password=&channelid=20d9884f-ae8c-45d3-ac5a-505ec258f01b&frameWidth=1920&frameHeight=1080&x=250&y=800&width=100&height=100
```

HTTP-интерфейс для получения медиа данных

Кроме возможности получения текстовой информации о системе и управления её компонентами HTTP-интерфейс **Macroscop** даёт возможность получать доступ в том числе и к такому содержимому, как кадры и видео от подключенных камер с помощью CGI-запросов.

Такие запросы в общем виде имеют следующий формат:

```
{Протокол}://{Сервер}:{Порт}/{Ресурс}?login={Логин}&password={Пароль}&channelid={Канал}&{Параметр}={Значение параметра}
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	8080	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https
Ресурс	–	URI ресурса сервера, к которому выполняется запрос
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос. Пользователь должен обладать правами доступа к каналам, функциям и возможностям системы, к которым обращается в рамках запроса
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
Канал	–	Номер, имя или уникальный идентификатор канала. Могут быть получены при выполнении запроса Получение конфигурации системы . Обязательный параметр
Параметр	–	Дополнительный параметр, уточняющий сам запрос или ответ на него. В зависимости от запроса может быть допустимо применение нескольких дополнительных параметров одновременно
Значение параметра	–	Значение применённого дополнительного параметра

Получение кадра

Для получения от сервера кадров для определённой камеры используется запрос к ресурсу **site**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
withcontenttype	false	Параметр, определяющий необходимость указать в ответе от сервера заголовок с типом передаваемых данных. Допустимые значения: true , false . Опциональный параметр

mode	realtime	Параметр, определяющий режим потока для передачи кадров. Допустимые значения: realtime — кадр из потока реального времени, archive — кадр из архива. При применении параметра mode=archive необходимо также задать параметр starttime . Опциональный параметр
starttime	–	Параметр, указывающий время в архиве, для которого необходимо передать кадры. Данный параметр состоит из комбинации даты и UTC-времени в формате dd.MM.yyyy HH:mm:ss или dd.MM.yyyy HH:mm:ss.fff . Обязательный параметр, если mode=archive .
resolution x	64	Желаемая ширина запрашиваемого кадра. Указывается в пикселях. Не может превышать действительную ширину кадра. Опциональный параметр
resolution y	36	Желаемая высота запрашиваемого кадра. Указывается в пикселях. Не может превышать действительную высоту кадра. Опциональный параметр
streamtype	Последнее доступное значение списка	Параметр, указывающий поток для получения кадров. Допустимые значения: <ul style="list-style-type: none"> • Main — поток Основной; • Alternative — поток Дополнительный 1; • SecondAlternative — поток Дополнительный 2; • ThirdAlternative — поток Дополнительный 3. Доступность указанных значений зависит от настроек канала

Пример запроса без дополнительных параметров:

```
http://127.0.0.1:8080/site?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f
```

В ответ на указанный запрос сервер вернёт кадр из потока реального времени для выбранной камеры.



По умолчанию кадр берётся из последнего по очередности потока. Например, при получении сервером от камеры четырёх потоков, кадр будет браться из потока «Дополнительный 3», что соответствует четвёртому потоку камеры.

Пример 1 запроса с дополнительными параметрами:

```
http://127.0.0.1:8080/site?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&streamtype=SecondAlternative
```

В ответ на этот запрос сервер вернёт кадр из потока Дополнительный 2, если это возможно.

Пример 2 запроса с дополнительными параметрами:

```
http://127.0.0.1:8080/site?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&withcontenttype=true&mode=archive&starttime=01.01.2023 00:00:01&resolutionx=640&resolutiony=480
```

В ответ на указанный запрос сервер вернёт кадр из архива, снятого 01 января 2023 года в 00:00:01 по часовому поясу UTC, добавив к ответу заголовок с типом данных и изменив разрешение на 640x480, если это возможно.

Получение «сырого» видео

Одним из способов получения потока видео от сервера **Macroscop** является запрос к ресурсу **video**, возвращающий в ответ полученные сервером от камеры данные без перекодирования.



Данный запрос предназначен в первую очередь для использования в сторонних приложениях, выполняющих роль **клиента**. Для просмотра полученных данных может потребоваться использование дополнительных декодирующих библиотек.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
channel	–	Имя канала. Может быть получено в интерфейсе приложений и при выполнении запроса Получение конфигурации системы . Обязательный параметр, если не используется channelid или channelnum
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр, если не используется channel или channelnum
channelnum	–	Номер канала в действующей конфигурации. Выясняется путём ручного пересчёта каналов в ответе на запрос Получение конфигурации системы . Обязательный параметр, если не используется channel или channelid
sound	off	Флаг получения видео- и аудиоданных канала в рамках одного соединения. Возможные значения: on – включить получение звука, off – выключить. При sound=on сервер возвращает звуковые кадры в формате G.711U, чередуя их в потоке данных с видеокадрами. Опциональный параметр
streamtype	Main	Параметр, указывающий поток для получения кадров. Допустимые значения: <ul style="list-style-type: none"> • Main — поток Основной; • Alternative — поток Дополнительный 1; • SecondAlternative — поток Дополнительный 2; • ThirdAlternative — поток Дополнительный 3. Доступность указанных значений зависит от настроек канала
mode	realtime	Параметр, определяющий режим потока для передачи кадров. Допустимые значения: realtime (поток реального времени), archive (архив). При применении параметра mode=archive необходимо также задать параметр starttime . Опциональный параметр
starttime	–	Параметр, указывающий время в архиве, для которого необходимо передать кадры. Данный параметр состоит из комбинации даты и UTC-времени в формате DD.MM.YYYY hh:mm:ss или DD.MM.YYYY hh:mm:ss.fff . Обязательный параметр при mode=archive
speed	1	Параметр, определяющий скорость воспроизведения потока. Допустимые значения: от 0.1 до 20. Опциональный параметр, применяется только при mode=archive



Параметры **channel**, **channelnum** и **channelid** являются взаимозаменяемыми, в связи с чем запрос должен содержать только один из них.

- Параметр **channel** передаёт в качестве значения имя канала в том виде, как оно представлено в действующей конфигурации. В случае существования в конфигурации нескольких каналов с одинаковым именем или при изменении имени канала в конфигурации могут возникнуть коллизии и ошибки из-за некорректного значения параметра, соответственно.
- Параметр **channelnum** передаёт в качестве значения порядковый номер канала в списке всех каналов действующей конфигурации. Использование данного параметра может вызвать трудности при работе с крупными системами. Помимо этого, порядковый номер канала может измениться при перемещении и удалении каналов из действующей конфигурации.
- Параметр **channelid** передаёт в качестве значения уникальный идентификатор канала (GUID), генерирующийся при создании канала и остающийся неизменным на протяжении всего времени существования канала в конфигурации.

Во избежание возможных проблем с выполнением запросов рекомендуется использовать для указания канала параметр **channelid**.



Все примеры запросов ниже составлены с использованием параметра **channelid**.

Пример запроса без дополнительных параметров:

```
http://127.0.0.1:8080/video?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f
```

В ответ на запрос сервер возвращает «бесконечный» HTTP-ответ, в котором содержатся видеокадры, разделённые заголовками.

Пример ответа на запрос:

```
HTTP/1.1 200 OK
...
Content-Type: multipart/x-mixed-replace; boundary=myboundary

-- myboundary
Content-Type: image/jpeg
Content-Length: 63125

<тело JPEG кадра>
```

При указании параметра **sound=on** видеокадры в «бесконечном» ответе чередуются с аудиокадрами.

Пример запроса с параметром **sound=on**:

```
http://127.0.0.1:8080/video?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&sound=on
```

Пример ответа на запрос с аудиокадром:

```
HTTP/1.1 200 OK
...
Content-Type: multipart/x-mixed-replace; boundary=myboundary
```

```
-- myboundary
Content-Type: audio, PCMU
Content-Length: 1000

<тело G711U кадра>
```

Для получения кадров из архива необходимо дополнить запрос соответствующими параметрами.

Пример запроса кадров из архива:

```
http://127.0.0.1:8080/video?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&mode=archive&startTime=01.01.2023 00:00:01&speed=3
```

В «бесконечном» ответе сервер начнёт передачу кадров из архива на скорости x3, начиная с кадра архива, снятого 01 января 2023 года в 00:00:01 по часовому поясу UTC. Структура ответа при этом полностью идентична ответам с кадрами реального времени.

Независимо от применённых параметров запроса, ответ от сервера всегда будет содержать следующие параметры:

Параметр	Описание
Content-Type	Заголовок MIME типа передаваемого кадра. В зависимости от заданного формата видео может принимать следующие значения: <ul style="list-style-type: none"> • image/jpeg — видеокادر, формат MJPEG. • video, mpeg4, I-frame — опорный кадр, формат MPEG4 • video, mpeg4, P-frame — разностный (промежуточный) кадр, формат MPEG4 • video, h264, I-frame — опорный кадр, формат H.264 • video, h264, P-frame — разностный (промежуточный) кадр, формат H.264. • audio, PCMU — аудиокادر, формат G.711U
Content-Length	Заголовок размера передаваемого кадра. Значение указывается в байтах.



При передаче опорных кадров форматов видео MPEG-4 и H.264, помимо информации о самом кадре также передаётся инициализирующая информация для декодера соответствующего формата.

Получение перекодированного видео в формате MJPEG

При обращении к ресурсу [video](#) сервер **Macroscop** возвращает видео в исходном (полученном от камеры) формате. Для некоторых приложений и непроизводительных устройств декодирование видео в формате H.264 или отображение MJPEG-видео в оригинальном разрешении может составить проблему.

Для таких случаев **Macroscop** способен отдавать видео- и аудиопотоки, перекодированные силами мобильного сервера. Получение перекодированных потоков возможно с использованием запроса к ресурсу **mobile**.

Дополнительные параметры запроса:

Параметр	Значение по	Описание
----------	-------------	----------

	умолчанию	
channel	–	Имя канала. Может быть получено в интерфейсе приложений и при выполнении запроса Получение конфигурации системы . Обязательный параметр, если не используется channelid или channelnum
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр, если не используется channel или channelnum
channelnum	–	Номер канала в действующей конфигурации. Выясняется путём ручного пересчёта каналов в ответе на запрос Получение конфигурации системы . Обязательный параметр, если не используется channel или channelid
sound	off	Флаг получения видео- и аудиоданных канала в рамках одного соединения. Возможные значения: on – включить получение звука, off – выключить. При sound=on сервер возвращает звуковые кадры в формате G.711U, чередуя их в потоке данных с видеокадрами. Опциональный параметр
streamtype	Main	Параметр, указывающий поток для получения кадров. Допустимые значения: <ul style="list-style-type: none"> • Main — поток Основной; • Alternative — поток Дополнительный 1; • SecondAlternative — поток Дополнительный 2; • ThirdAlternative — поток Дополнительный 3. Доступность указанных значений зависит от настроек канала
soundformat	g711u	Указание формата для перекодирования потока аудио. Возможные значения: pcm , g711u , g711a , aac . Опциональный параметр
fps	Частота потока мобильного сервера	Желаемая частота кадров в секунду. Фактическая частота может отличаться от запрошенной, т.к. зависит от множества параметров. Максимальная частота ограничена настройками перекодированного потока для мобильного сервера. Опциональный параметр
oneframeonly	false	Флаг запроса одиночного кадра с последующим завершением соединения. Может использоваться как альтернатива запросу site . Возможные значения: true , false . Опциональный параметр
resolutionx	Значение Плохого качества	Желаемая ширина кадра. Используется для подбора наиболее подходящего уровня качества для перекодированного потока. Опциональный параметр
resolutiony	Значение Плохого качества	Желаемая высота кадра. Используется для подбора наиболее подходящего уровня качества для перекодированного потока. Опциональный параметр
mode	realtime	Параметр, определяющий режим потока для передачи кадров. Допустимые значения: realtime (поток реального времени), archive (архив). При применении параметра mode=archive необходимо также задать параметр starttime . Опциональный параметр
starttime	–	Параметр, указывающий время в архиве, для которого необходимо передать кадры. Данный параметр состоит из комбинации даты и UTC-времени в формате DD.MM.YYYY

		hh:mm:ss или DD.MM.YYYY hh:mm:ss.fff . Обязательный параметр при mode=archive
isforward	true	Направление воспроизведения архива. Возможные значения: true – архив проигрывается в хронологическом порядке, false – в обратном хронологическом порядке. Опциональный параметр, применяется только при mode=archive
speed	1	Параметр, определяющий скорость воспроизведения потока. Допустимые значения: от 0.1 до 20. Опциональный параметр, применяется только при mode=archive
withcontenttype	false	Параметр, определяющий необходимость указать в ответе от сервера заголовок с типом передаваемых данных. Допустимые значения: true , false . Опциональный параметр



При работе с запросом **mobile** необходимо учитывать, что под термином «поток» могут пониматься сразу две сущности:

- Исходный поток видео
- Перекодированный мобильным сервером поток

Под исходным потоком в данном случае понимается поток, получаемый от камеры сервером **Macroscop** под названием **Основной** (Main), **Дополнительный 1** (Alternative), **Дополнительный 2** (SecondAlternative) или **Дополнительный 3** (ThirdAlternative). Такой поток выступает в качестве источника кадров для перекодирования мобильным сервером и указывается в параметре **streamtype**.

Под перекодированным потоком понимается тот же исходный поток, но уже преобразованный мобильным сервером в соответствии с собственными настройками. Именно его передаёт сервер в ответ на запрос.



Запрос **mobile** для передачи потоков использует механизмы перекодирования мобильного сервера. В связи с этим получаемый поток подчиняется ограничениям, накладываемым настройками мобильного сервера.

Мобильный сервер предоставляет возможность задать до трёх уровней качества перекодируемого потока: **Плохого**, **Среднего** и **Хорошего**. Для каждого уровня качества можно задать собственные настройки разрешения кадра и их частоту в потоке. Для исходных потоков форматов H.264, H.265, MPEG4 и MxPEG также можно задать перекодирование с использованием только опорных кадров.

При выполнении запроса **mobile** сервер автоматически выбирает наиболее близкий к указанным значениям уровень качества и применяет настройки этого уровня к перекодируемому потоку: разрешение уровня в качестве разрешения потока, частота кадров уровня в качестве максимальной частоты кадров потока.

Пример 1:

Запрос включает в себя параметры resolutionx=720, resolutiony=500 и fps=20. Ближайший по параметрам уровень качества **Среднее** имеет разрешение 640x480 и fps=10. Полученный перекодированный поток будет полностью соответствовать уровню качества **Среднее**.

Пример 2:

Запрос включает в себя параметры resolutionx=600, resolutiony=460 и fps=5. Ближайшим по параметрам уровнем качества вновь является **Среднее** с теми же настройками. Полученный перекодированный поток в таком случае будет иметь разрешение 640x480 и fps=5.

Узнать действующие настройки мобильного сервера можно с помощью запроса на [Получение конфигурации системы](#) – параметры перекодирования перечислены в разделе **MobileServerInfo**.

Изменить настройки мобильного сервера можно в разделе [Мобильные устройства](#) настроек сервера в приложении **Macroscop Конфигуратор**.



Мобильный сервер использует одни и те же настройки для передачи потоков в мобильные приложения, в Web-Клиент и в ответ на запрос mobile. Изменение настроек мобильного сервера может негативно повлиять на других пользователей системы.

Мобильный сервер имеет дополнительные настройки прав доступа пользователей. Пользователю, отправляющему запрос, должно быть предоставлено право на **Подключение с мобильных устройств и Web-Клиента**.

Узнать действующие настройки права можно с помощью запроса на [Получение конфигурации системы](#) – параметр **CanGetTranscodedVideoFromMobileServer** раздела **UserGroup**.

Изменить настройки прав можно в разделе [Пользователи](#) приложения **Macroscop Конфигуратор**.



Параметры **channel**, **channelnum** и **channelid** являются взаимозаменяемыми, в связи с чем запрос должен содержать только один из них.

- Параметр **channel** передаёт в качестве значения имя канала в том виде, как оно представлено в действующей конфигурации. В случае существования в конфигурации нескольких каналов с одинаковым именем или при изменении имени канала в конфигурации могут возникнуть коллизии и ошибки из-за некорректного значения параметра, соответственно.
- Параметр **channelnum** передаёт в качестве значения порядковый номер канала в списке всех каналов действующей конфигурации. Использование данного параметра может вызвать трудности при работе с крупными системами. Помимо этого, порядковый номер канала может измениться при перемещении и удалении каналов из действующей конфигурации.
- Параметр **channelid** передаёт в качестве значения уникальный идентификатор канала (GUID), генерирующийся при создании канала и остающийся неизменным на протяжении всего времени существования канала в конфигурации.

Во избежание возможных проблем с выполнением запросов рекомендуется использовать для указания канала параметр **channelid**.



Все примеры запросов ниже составлены с использованием параметра **channelid**.

Пример запроса 1:

```
http://127.0.0.1:8080/mobile?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f
```

В ответ сервер начнёт передачу потока в соответствии с настройками самого низкого уровня качества из настроенных.

Пример запроса 2:

```
http://127.0.0.1:8080/mobile?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&resolutiony=480&sound=on
```

В ответ сервер автоматически определит наиболее близкий по значениям настроек уровень качества и начнёт передачу потока в соответствии с его настройками.

Пример запроса 3:

```
http://127.0.0.1:8080/mobile?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&resolutiony=480&oneframeonly=true&mode=archive
&starttime=01.01.2023 00:00:01
```

В ответ сервер автоматически определит наиболее близкий по значениям настроек уровень качества и выполнит передачу одиночного кадра из архива, снятого 01 января 2023 года в 00:00:01, после чего завершит сеанс передачи.



Если планируется воспроизводить получаемый перекодированный поток в браузере, необходимо задать для параметра **withcontenttype** значение **true**.

Получение фрагмента видео из архива в формате MP4

Помимо получения потокового вещания видео- и аудиоданных, сервер **Macroscop** также предоставляет возможность экспортировать фрагменты архива в виде файлов в формате MP4. Для этого используются запросы к ресурсу **exportarchive**.



Функция поддерживается только для камер, архив которых записывается в форматах H.264, H.265 и MJPEG.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
channelid	–	Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
fromtime	–	Дата и время начала периода, за который запрашивается информация. Время должно указываться в часовом поясе UTC в формате DD.MM.YYYY hh:mm:ss. Обязательный параметр.
totime	–	Дата и время конца периода, за который запрашивается информация. Время должно указываться в часовом поясе UTC в формате DD.MM.YYYY hh:mm:ss. Обязательный параметр
sound	off	Параметр для добавления аудиоданных из архива в экспортируемый файл. Доступно только для серверов под управлением Windows. Возможные значения: on – экспортировать звук, off – не экспортировать. Опциональный параметр
usetimestamps	false	Параметр, позволяющий накладывать поверх видео временную метку с указанием времени записи

		отображаемого кадра. Возможные значения: true – накладывать временную метку, false – не накладывать. Опциональный параметр
fromDevice	false	Параметр запроса на экспорт архива, хранящегося непосредственно на камере или видеорегистраторе, если они поддерживают функцию доступа к архиву. Возможные значения: true – экспортировать архив с устройства, false – экспортировать архив с сервера. Опциональный параметр
addhvc1tagforhvc	false	Параметр для добавления заголовка, позволяющего воспроизведение экспортированного фрагмента формата H.265 на устройствах Apple. Незначительно увеличивает продолжительность экспорта. Возможные значения: true – добавить заголовок, false – не добавлять. Опциональный параметр

Пример запроса:

```
http://127.0.0.1:8080/exportarchive?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&fromtime=01.01.2023 00:00:00&totime=01.01.2023 00:30:00&sound=on&usetimestamps=true
```

В ответ на такой запрос сервер подготовит файл в формате MP4, содержащий фрагмент архива с аудиодорожкой и временными метками, записанный 01 января 2023 года в период с 00:00:00 до 00:30:00. Скачивание файла начнётся автоматически, если соединение с сервером не будет прервано закрытием приложения.



Запрос имеет ряд технических ограничений, которые необходимо учитывать при использовании:

- Запросы на экспорт фрагмента архива при получении добавляются в очередь сервера на выполнение. Максимальный размер очереди – 10 запросов.
- Максимальное количество одновременно обрабатываемых запросов – 5. Таким образом, при полной загрузке очереди обрабатываться будут только 5 запросов, поступившие первыми, пока остальные 5 будут находиться в режиме ожидания.
- Все запросы сверх максимального размера очереди будут отброшены с кодом сообщения 500.
- Максимальная продолжительность экспортируемого фрагмента – 1 час.

HTTP-интерфейс для управления автосменой видов

HTTP-интерфейс позволяет задавать профиль автосмены и получать информацию о ней.

Пример запроса:

```
{Протокол}://{Сервер}:{Порт}/{Ресурс}?type={Тип}&{Параметр}={Значение параметра}&login={Логин}&password={Пароль}
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	8080	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https
Ресурс	–	URI ресурса сервера, к которому выполняется запрос
Тип	–	Тип запроса, может принимать следующие значения: setautoswitchview и getautoswitchviews
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос. Пользователь должен обладать правами доступа к каналам, функциям и возможностям системы, к которым обращается в рамках запроса
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
Параметр	–	Дополнительный параметр, уточняющий сам запрос или ответ на него. В зависимости от запроса может быть допустимо применение нескольких дополнительных параметров одновременно
Значение параметра	–	Значение применённого дополнительного параметра

Установка профиля автосмены видов

Для установки профиля автосмены используется тип запроса **setautoswitchview**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
clientip	–	IP адрес устройства, на котором запущено клиентское приложение Macroscop . Обязательный параметр
monitor	–	Номер монитора в действующей конфигурации клиентского приложения. Отсчёт начинается с 0 . Обязательный параметр

autoswitch viewid	—	Уникальный идентификатор, созданный администратором в конфигураторе Macroscop , используется для создания профиля автосмены видов и применения его в клиенте Macroscop
------------------------------	---	--

Пример запроса:

```
http://127.0.0.1:8080/command?type=setautoswitchview&clientip=127.0.0.1&monitor=0&autoswitchviewid=d50ce608-978e-4a49-b0f6-4406ae7f2975&login=root&password
```

Получение профилей автосмены видов

Для установки профиля автосмены используется тип запроса **setautoswitchview**.

Дополнительные параметры запроса:

Параметр	Значение по умолчанию	Описание
responsetype	xml	Формат представления возвращаемых данных. Если не указан в запросе, используется значение по умолчанию. Опциональный параметр. Допустимые значения: xml, json

Пример запроса:

```
http://127.0.0.1:8080/command?login=root&type=getautoswitchviews&password=&responsetype=json
```


Пример ответа в JSON-формате:

```
[
  {
    "Id": "d50ce608-978e-4a49-b0f6-4406ae7f2975",
    "Name": "Новая автосмена 1" },
  {
    "Id": "ae491b1a-8d3b-4031-a3fc-d84b94781ebe",
    "Name": "Новая автосмена 2"}
]
```

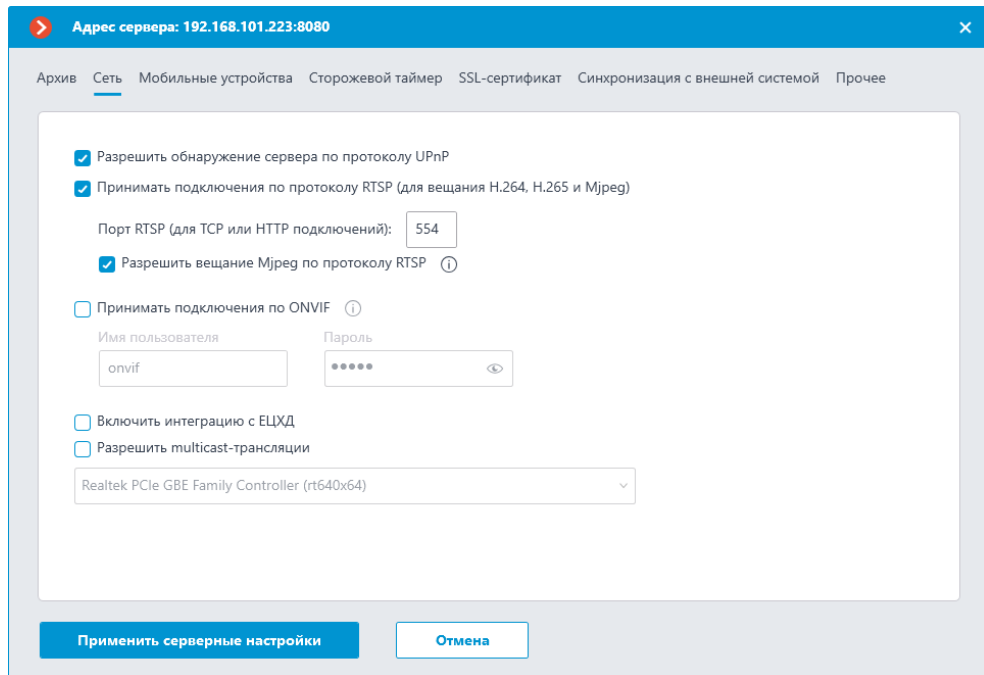
Примечание: В HTTP-запросах на выполнение команд в клиенте **Macroscop** добавлена переменная **hostname**, позволяющая задавать адрес клиента по имени хоста и IP-адресу. Переменная **clientip**, аналогичная по функции с **hostname**, сохранена для обратной совместимости. В случае одновременного использования переменных в запросе, будет использоваться только значение **hostname**.

RTSP-интерфейс для получения видео и звука


RTSP-интерфейс используется для получения видео и звука клиентами, работающими по протоколу RTSP. Данный интерфейс поддерживает кодек H.264 и, опционально, MJPEG (MJPEG по умолчанию выключен).

 Альтернативный способ формирования URL для обращения к RTSP-интерфейсу описан на странице [Генерация RTSP-ссылок на камеры в приложении Macroscop Конфигуратор](#) руководства администратора.

Перед началом использования следует удостовериться, что RTSP-интерфейс включен. Для этого нужно запустить **Macroscop Конфигуратор** и на странице серверных настроек в разделе **Сеть** убедиться, что установлен флажок **Принимать подключения по протоколу RTSP (для вещания H.264 и MJPEG)**. В том же блоке настроек указан порт для RTSP подключений.



Для подключения по протоколу RTSP можно использовать TCP-подключения (RTSP over TCP) или HTTP-подключения (RTSP over HTTP). UDP-подключения (RTSP over UDP) не поддерживаются.

 По умолчанию вещание формата MJPEG по протоколу RTSP отключено, поскольку протокол RTSP поддерживает только MJPEG-кадры, закодированные в базовом (Baseline) режиме кодирования. Таким образом, для передачи видеопотоков, закодированных в других режимах MJPEG, потребуется перекодирование; что, в свою очередь, повысит нагрузку на сервер. Кроме того, при перекодировании MJPEG может быть понижена частота кадров (по сравнению с частотой кадров, передаваемой непосредственно камерой).

Подключение к серверу осуществляется RTSP-клиентом, например VLC, с помощью строки подключения следующего вида:

```
rtsp://{Сервер}:{Порт}/rtsp?login={Логин}&password={Пароль}&{Канал}={Значение канала}&{Параметр}={Значение параметра}
```

Где:

Параметр	Значение по умолчанию	Описание
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	554	Сетевой порт протокола RTSP
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос. Пользователь должен обладать правами доступа к каналам, функциям и возможностям системы, к которым обращается в рамках запроса
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
Канал	–	Способ указания канала: channelid или channelnum
Значение канала	–	GUID канала для channelid или номер канала для channelnum
Параметр	–	Дополнительный параметр, уточняющий сам запрос или ответ на него. В зависимости от запроса может быть допустимо применение нескольких дополнительных параметров одновременно
Значение параметра	–	Значение применённого дополнительного параметра



Параметры **channelid** и **channelnum** являются взаимозаменяемыми, в связи с чем запрос должен содержать только один из них. **channelnum** использует в качестве значения порядковый номер канала в списке каналов конфигурации. Любое изменение конфигурации может привести к изменению номера, что может повлечь ошибки при выполнении запроса.

Рекомендуется использовать **channelid**, использующий неизменяемый GUID камеры, сгенерированный при создании канала.

RTSP-запрос также может содержать дополнительные параметры, предоставляющие дополнительные возможности:

Параметр	Значение по умолчанию	Описание
sound	off	Флаг получения видео- и аудиоданных канала в рамках одного соединения. Возможные значения: on – включить получение звука, off – выключить. Опциональный параметр
streamtype	Main	Параметр, указывающий поток для получения кадров. Допустимые значения: <ul style="list-style-type: none"> • Main — поток Основной; • Alternative — поток Дополнительный 1; • SecondAlternative — поток Дополнительный 2; • ThirdAlternative — поток Дополнительный 3. Доступность указанных значений зависит от настроек канала
mode	realtime	Параметр, определяющий режим потока для передачи кадров. Допустимые значения: realtime (поток реального времени), archive (архив). При применении

		параметра mode=archive необходимо также задать параметр starttime . Опциональный параметр
starttime	–	Параметр, указывающий время в архиве, для которого необходимо передать кадры. Данный параметр состоит из комбинации даты и UTC-времени в формате DD.MM.YYYY hh:mm:ss или DD.MM.YYYY hh:mm:ss.fff . Обязательный параметр при mode=archive
speed	1	Параметр, определяющий скорость воспроизведения потока. Допустимые значения: от 0.1 до 20. Опциональный параметр, применяется только при mode=archive
isforward	true	Направление воспроизведения архива. Возможные значения: true – архив проигрывается в хронологическом порядке, false – в обратном хронологическом порядке. Опциональный параметр, применяется только при mode=archive

Пример запроса 1:

```
rtsp://127.0.0.1:554/rtsp?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f
```

В ответ сервер начнёт передачу видеок кадров из потока в режиме реального времени для указанного канала.

Пример запроса 2:

```
rtsp://127.0.0.1:554/rtsp?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&sound=on
```

В ответ сервер начнёт передачу видео- и аудиока кадров из потока в режиме реального времени для указанного канала.

Пример запроса 3:

```
rtsp://127.0.0.1:554/rtsp?login=root&password=&channelid=706c4691-3d90-41e3-8789-76eb9810648f&sound=on&mode=archive&starttime=01.01.2023 23:59:59&isforward=false&speed=2
```

В ответ сервер начнёт передачу видео- и аудиока кадров из архива указанного канала за 01 января 2023 года, начав воспроизведение в обратном хронологическом порядке со скоростью x2 с 23:59:59.

Macroscop API с интерфейсом XML



Для работы с XML-интерфейсом системы необходимо использовать приложения, способные отправлять на указанный URL тело запроса в формате XML.

XML интерфейс позволяет посылать на сервер **Macroscop** запросы в формате XML и получать в ответ данные в том же формате. Для того, чтобы выполнить запрос данных в формате XML, необходимо обратиться к следующему URL:

```
{Протокол}://{Сервер}:{Порт}/xml
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	–	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https

В теле запроса необходимо указать основные параметры запроса согласно следующей структуре:

```
<?xml version="1.0" encoding="utf-8" ?>
<query>
  <server_login>root</server_login>
  <server_pass_hash></server_pass_hash>
  <query_name>get_people_counters</query_name>
  <query_params>
    ...
  </query_params>
</query>
```

Ниже описано назначение параметров тела запроса:

Параметр	Значение по умолчанию	Описание
server_login	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос
server_pass_hash	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, параметр можно оставить пустым или не указывать в запросе
query_name	–	Строковое наименование типа запроса
query_params	–	Параметр-контейнер для указания параметров, специфичных для указанного в query_name типа запроса

В ответ сервер возвращает ответ вида:

```
<?xml version="1.0" encoding="utf-8" ?>
<result>
  <query_name></query_name>
  <query_result>0k</query_result>
  <query_msg>Запрос выполнен успешно.</query_msg>
```

```
<query_time>20.09.2012 10:58:15</query_time>
<query_time_local>20.09.2012 16:58:15</query_time_local>
</result>
```

Где:

Параметр	Описание
query_name	Строковое наименование типа запроса
query_result	Результат выполнения запроса: Ok — если запрос выполнен успешно Error — если произошли какие-либо ошибки
query_msg	Строковый комментарий по результатам выполнения запроса
query_time	Время запроса UTC
query_time_lo cal	Время запроса локальное



В ответе также могут содержаться другие параметры, специфичные для указанного типа запроса.

Получение данных счётчика посетителей

Для получения данных счётчика посетителей используется тип запроса **get_people_counters**.

Параметрами данного запроса являются:

Параметр	Значение по умолчанию	Описание
channel_id	–	Идентификатор канала, на котором настроен счётчик посетителей
search_time	–	Момент времени, для которого необходимо вернуть показания счётчика. Время указывается в формате YYYY-MM-dd hh:mm:ss. При этом указывается локальное время сервера, на который отправляется запрос

Пример тела запроса:

```
<?xml version="1.0" encoding="utf-8" ?>
<query>
  <server_login>root</server_login>
  <server_pass_hash></server_pass_hash>
  <query_name>get_people_counters</query_name>
  <query_params>
    <channel_id>cacdd8e6-1c56-435c-86e3-6967d7494a50</channel_id>
    <search_time>2012-09-17 09:50:00</search_time>
  </query_params>
</query>
```

В ответ сервер возвращает следующие параметры:

```
<in>434</in>  
<out>378</out>
```

Где:

Параметр	Описание
in	Количество вошедших людей для данного счётчика
out	Количество вышедших людей для данного счётчика

Организация вещания видео на сайт

Вещание видео на сайт может быть организовано с помощью службы мобильных подключений сервера **Macroscop** и компонентов на клиентской стороне (в браузере).

Вещание с помощью HTML5

 Реализовано в Macroscop версии 4.0 и более поздних.

Вещание видео в режиме реального времени на сайт может быть организовано с помощью службы мобильных подключений сервера Macroscop и HTML5-проигрывателя, предоставляемого Web-клиентом Macroscop.

Для организации вещания необходимо сформировать ссылку на поток выбранной камеры следующего вида:

```
{Протокол}://{Сервер}:{Порт}/embedding/index.html#/embed?login={Логин}&password={Пароль}&channelid={Канал}&channelstreamtype={Поток}&mode={Формат}
```

Где:

Параметр	Значение по умолчанию	Описание
Протокол	http	Сетевой протокол, выбранный для взаимодействия с сервером Macroscop . По умолчанию используется http , доступность https определяется настройками сервера
Сервер	–	Доменное имя или IP адрес сервера Macroscop
Порт	–	Сетевой порт, соответствующий выбранному Протоколу . Порты по умолчанию: 8080 для http ; 18080 для https
Логин	–	Имя пользователя Macroscop , от имени которого будет выполняться запрос
Пароль	–	md5-хэш пароля пользователя Macroscop . Если пароль для пользователя не задан, в качестве значения параметра необходимо указать md5-хэш пустой строки (D41D8CD98F00B204E9800998ECF8427E)
Канал		Уникальный идентификатор канала. Может быть получен при выполнении запроса Получение конфигурации системы . Обязательный параметр
Поток	–	Тип потока, предпочтительного для вещания. Может принимать значения: Main (Основной), Alternative (Альтернативный 1), SecondAlternative (Альтернативный 2), ThirdAlternative (Альтернативный 3)
Формат	–	Предпочитаемый формат видео: MJPEG или H264



Регистрозависимым является только параметр **Логин**, в то время как остальные параметры могут быть указаны в любом регистре.

Пример ссылки:

```
https://188.17.220.37:18080/embedding/index.html#/embed?login=Website&password=5989aeb826edde08a1109deb5e61a4ba&channelid=d8112e29-fce9-40ea-bef4-a1c7b276ac98&channelstreamtype=Alternative&mode=H264
```

Доступ к потоку видео осуществляется с помощью компонентов **Web-клиента Macroscop**, в связи с чем необходимо выполнить следующие настройки в приложении **Macroscop Конфигуратор**:

- Поток должен быть разрешён для использования мобильными приложениями и Web-клиентом. Для этого необходимо в разделе **Камеры**, выбрать нужную камеру и убедиться, что у выбранного для вещания потока включена опция **Мобильные и веб клиенты**.
- Пользователь, от чьего имени запрашивается поток, должен иметь право на использование встраиваемого компонента. Для этого необходимо в разделе **Пользователи** выбрать группу пользователя и нажать **Редактировать**. В открывшемся окне, на вкладке **Основные** необходимо убедиться, что право **Подключение с мобильных устройств и Web-Клиента** включено.
- Пользователь, от чьего имени запрашивается поток, должен иметь право на просмотр выбранной для вещания камеры. Для этого необходимо в разделе **Пользователи** выбрать группу пользователя и нажать **Редактировать**. В открывшемся окне, на вкладке **Камеры** необходимо убедиться, что для выбранной камеры включено право **Наблюдение** или **Наблюдение и архив**.



В целях безопасности рекомендуется использовать для вещания видео на сайт отдельного пользователя, предоставив ему необходимый для этих целей минимум прав.

Предоставить доступ к видео по сформированной ссылке можно несколькими способами. Например, разместить на странице сформированную ссылку на поток как гиперссылку для открытия отдельной **вкладки с проигрывателем** или добавить в качестве содержимого **iframe-элемента** для воспроизведения непосредственно на странице.

Чтобы открывать проигрыватель в виде отдельной вкладки, необходимо разместить на странице элемент гиперссылки **<a>**, добавив в него ранее сформированную ссылку на поток:


```
<a href="https://188.17.220.37:18080/embedding/index.html#/embed?login=Website&password=5989aeb826edde08a1109deb5e61a4ba&channelid=d8112e29-fce9-40ea-bef4-a1c7b276ac98&channelstreamtype=Alternative&mode=H264">Просмотр камеры Парковка</a>
```



При открытии проигрывателя в виде отдельной вкладки ширина и высота изображения определяются шириной и высотой открывшейся вкладки, несмотря на исходные параметры получаемого потока.

Чтобы разместить проигрыватель непосредственно на странице, необходимо добавить в код страницы элемент встраиваемого содержимого **<iframe>**, добавив в него ранее сформированную ссылку на поток:

```
<iframe src="https://188.17.220.37:18080/embedding/index.html#/embed?login=Website&password=5989aeb826edde08a1109deb5e61a4ba&channelid=d8112e29-fce9-40ea-bef4-a1c7b276ac98&channelstreamtype=Alternative&mode=H264" frameborder="0" width="{Ширина}" height="{Высота}" allowfullscreen></iframe>
```

-  При встраивании проигрывателя в качестве содержимого `iframe`-элемента размеры изображения определяются заданными параметрами **Ширина** и **Высота**, несмотря на исходные параметры получаемого потока.

Описанный способ имеет ряд ограничений, которые необходимо учитывать при использовании:

- Для данного способа необходима поддержка [Media Source API](#) со стороны браузера. Актуальные версии браузеров, кроме **Safari для iOS**, [поддерживают этот API](#).
- Вещание доступно **только** для видео в режиме **реального времени**. Доступ к архиву не поддерживается.
- Вещание доступно **только** для **видео**. Получение звука от камеры данным способом не поддерживается.
- Вещание доступно **только** для потоков в формате **H264** и **MJPEG**. Другие форматы видео не поддерживаются.
- Вещание в формате **H264** доступно только при условии, что исходный **поток** транслируется **в том же формате**. Перекодирование исходного потока MJPEG в H264 не предусмотрено.
- Вещание в **MJPEG** доступно как для камер, изначально передающих **поток в этом формате**, так и для **потоков H264** путём **перекодирования** исходного изображения на сервере.
- При возникновении проблем с отображением потока в формате H264 произойдёт **автоматическая смена формата на MJPEG** без возможности обратного переключения со стороны пользователя.



Перекодирование H264 в MJPEG потребляет ресурсы сервера и может привести к повышенной нагрузке на сервер.

В случае возникновения сложностей с вещанием необходимо убедиться, что:

- Указанные **сервер**, **камера** и выбранный для вещания **поток** активны и **доступны** для использования.
- Пользователю **предоставлены** необходимые для получения потока **права**.
- Все обязательные **параметры** ссылки-запроса (данные пользователя, идентификатор камеры, параметры потока) **заполнены** корректными **данными**.



Подробную информацию о причинах можно получить в Консоли браузера. Для этого откройте сформированную ссылку как отдельную вкладку, запустите Инструменты разработчика (F12) и перейдите на вкладку Консоль.

Вещание с помощью Flash (устарело)



Данный способ является устаревшим в связи с прекращением поддержки технологии Flash со стороны разработчика Adobe.

Вещание видео на сайт может быть организовано с помощью службы мобильных подключений сервера Macroscop и Flash-компонента на клиентской стороне. Пример использования компонента на HTML-странице размещен в папке **Examples\SiteFlash** примеров плагинов SDK. На HTML-странице (**index.html**) необходимо указать параметры подключения к серверу Macroscop, а также идентификатор (имя/номер) канала, с которого должно транслироваться видео и требуемый кодек (H.264 или MJPEG).

Пример настройки:

```
var flashvars = {
    server: "demo.macroscop.com", // адрес сервера
    port: "8080", // порт сервера
    login: "root", // имя пользователя
    password_hash: "", // md5-хэш пароля
```

```

mode: "MJPEG," // предпочитаемый формат видео
channel: "1" // имя, номер или идентификатор канала
};

```

Идентификаторы всех каналов в системе могут быть получены с помощью соответствующего запроса (см. [Получение конфигурации системы](#)).

Параметр **Предпочитаемый формат видео (mode)** может принимать значения **MJPEG**, **H264** или вообще пропущен. Если предпочитаемый формат видео не задан, то автоматически будет выбран подходящий формат. Значение **H264** можно указать только для камер, транслирующих видеопотоки, закодированные в H.264. Значение **MJPEG** можно указать для всех камер, однако в случае перекодирования из H.264 это может привести к повышенной нагрузке на сервер.

Вещание с помощью JavaScript (устарело)



Данный способ является устаревшим, поскольку создает повышенную нагрузку на сервер Macroscop и предоставляет худшее качество по сравнению с другими способами.

Вещание видео на сайт может быть организовано с помощью сервера Macroscop и JavaScript-компонента на клиентской стороне. Скрипт для клиентской стороны и пример его использования на HTML-странице размещен в папке **Examples\Site\frameReceiver.js** примеров плагинов SDK. В скрипте необходимо указать параметры подключения к серверу Macroscop, а также идентификатор (имя/номер) канала, с которого должно транслироваться видео и требуемый размер области, в которую будут выводиться видеокдры.

Пример настройки скрипта:

```

var serverUrl = "http://95.23.84.1:8080" /*URL сервера*/
var login = "root" /*пользователь, имеющий права на просмотр транслируемого канала*/
var password = ""; /*MD5-хэш пароля пользователя в верхнем регистре или пуста строка, если пароль пустой*/
var channelnum = 0; /*порядковый номер канала в общей конфигурации, счет с 0*/
var drawWidth = 577; /*ширина области отображения, в пикселях*/
var drawHeight = 432; /*высота области отображения, в пикселях*/

```

Пример скрипта, использующего идентификатор канала вместо его порядкового номера, размещен в папке **Examples\Site\frameReceiver_id.js** примеров плагинов SDK.

На самой HTML-странице должен быть размещен тег ``, в котором будет отображаться видеопоток, закодированный в MJPEG.



Не рекомендуется изменять размеры области отображения видео динамически, поскольку это приведет к существенному повышению потребляемых ресурсов службой мобильных подключений, так как эта служба перекодирует в MJPEG исходный видеопоток, после чего полученный поток кадров разделяет между многими клиентами (сессиями). Использование разных разрешений также приведет к дополнительной нагрузке на службу мобильных подключений.