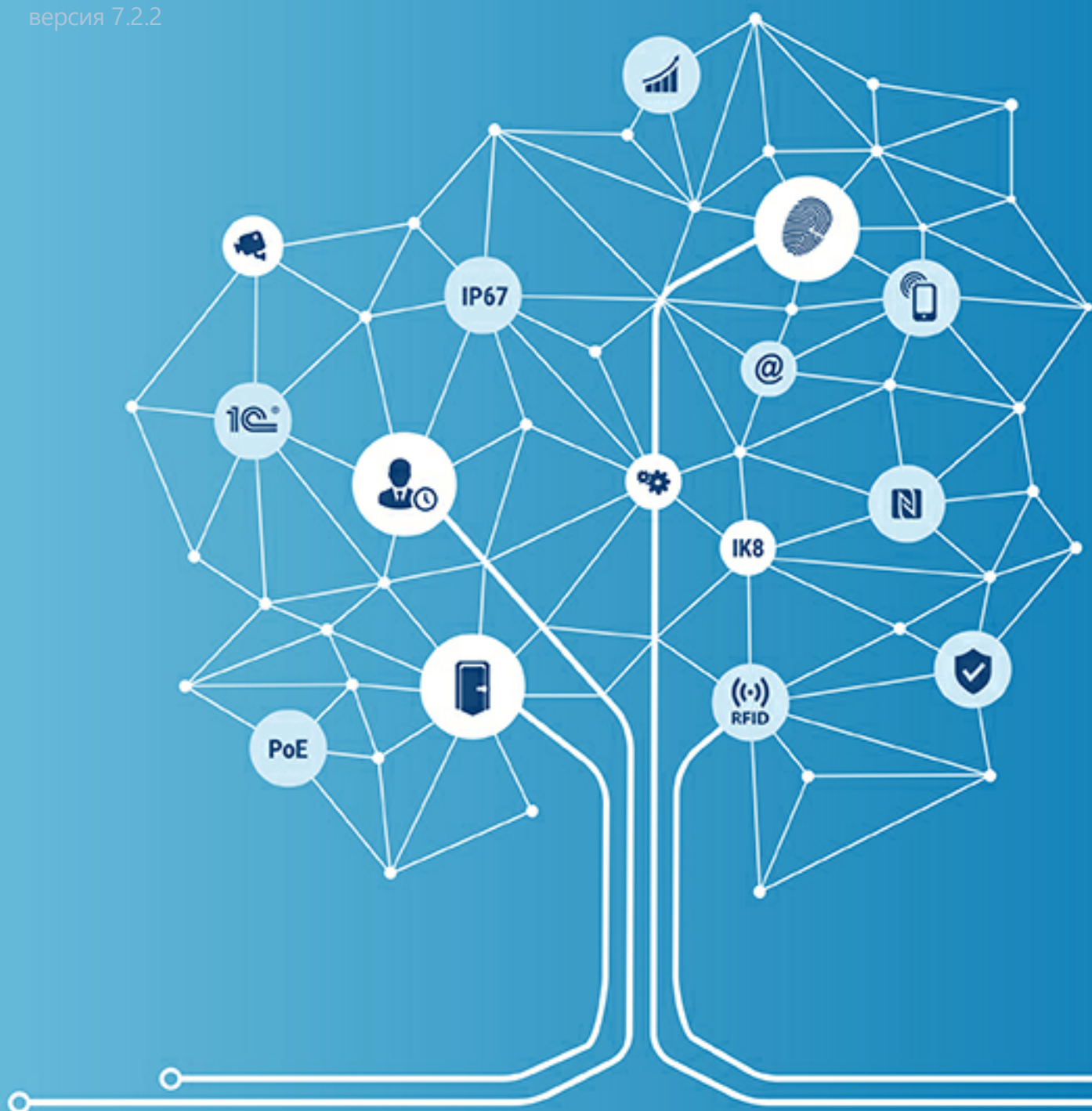




# Подсистема автоматизации и COM SDK

Руководство администратора

версия 7.2.2



## Соглашения, используемые в книге

В этой книге используются следующие соглашения о шрифтах:

- *курсив* — используется при ведении новых терминов и указаний текстовых значений переменных,
- **полужирное начертание** — используется для выделения названий элементов окон,
- ***полужирный курсив*** — используется для выделения названий окон,
- «кавычки» — используются для выделения названий приложений и пунктов меню.

## Оглавление

<b>1</b>	<b>Подсистема автоматизации и COM SDK</b>	<b>5</b>
<b>1.1</b>	<b>Подсистема автоматизации</b>	<b>5</b>
	<b>Конфигурирование подсистемы автоматизации</b>	<b>7</b>
	<b>Служба «Автоматизация»</b>	<b>9</b>
	Настройки автоматизации	10
	Макрокоманда VBScript	10
	Редактор макрокоманд VBScript	10
	Простая макрокоманда	11
	Программа VBScript	12
	Простая программа	12
	Фильтр событий	12
	Вкладка «Основные»	13
	Вкладка «Инициатор сообщения»	14
	Вкладка «Общие поля сообщений»	15
<b>1.2</b>	<b>Объектная модель, доступная из макрокоманд и программ VBScript</b>	<b>17</b>
	<b>Получение доступа к объектам и сообщениям из скрипта</b>	<b>18</b>
	<b>Предопределенные функции в программах VBScript</b>	<b>19</b>
	Функция Apacs_Event(PropEvt)	19
	Функция Script_Load()	20
	Функция Script_Close()	20
	<b>Предопределенные объекты в программах и макрокомандах VBScript</b>	<b>20</b>
	Интерфейс IApServerWrap	22
	Метод getObjectByAlias	22
	Метод getRootObject	23
	Метод getObjectByUID	24
	Метод getObjectsByFilter	25
	Метод getEvents	26
	Метод set_onEvent	27
	Метод set_onNotifyAdd	28
	Метод set_onNotifyChange	28
	Метод set_onNotifyDelete	29
	Интерфейс IApLogWrap	30
	Метод logBAD	30
	Метод logNORM	31
	Метод logTRK	31
	Интерфейс IApClientWrap	31
	Метод playSound	31
	Метод show Video	32
	Метод hideVideo	32
	Метод writeToLog	32
	Интерфейс IApScriptWrap	32
	Метод sleep	33
	Метод setTimeout	34
	Метод clearTimeout	34
	Метод setInterval	34
	Метод clearInterval	35
	Метод toVBAArray	36
	Метод toCPPArray	36
	Метод loadCPPArray	37
	Метод saveCPPArray	38
	Интерфейс IApObjectWrap	39
	Метод <команда>	40
	Метод getCurrentSettings	40
	Метод applySettings	41
	Метод getChildrenObjs	42
	Метод getChildrenObjsByTypes	43
	Метод getParentObject	44
	Метод deleteObject	44
	Метод getChildSettingsForAdd	45

Метод addChildWithSettings.....	46
Метод getEventSettingsForRegister.....	47
Метод registerEventWithSettings.....	48
Метод getUID.....	49
Метод getApacsType.....	49
Интерфейс IApcVideoLinkerWrap.....	50
Метод linkRec.....	50
Интерфейс IApcEQUALObjFilter.....	51
Интерфейс IApcANDObjFilter.....	51
Метод addCondition.....	51
Метод getConditions.....	52
Интерфейс IApcHelper.....	52
Метод setServer.....	52
Метод getAccount.....	53
Метод applyAccount.....	53
Метод getCardHolder.....	53
Метод applyCardHolder.....	53
Метод getSecurityGroup.....	53
Метод applySecurityGroup.....	53
<b>1.3 Клиентский модуль HTML обозреватель.....</b>	<b>53</b>
Редактор HTML обозревателя.....	55
<b>1.4 COM SDK APACS 3000 (комплект для разработки программного обеспечения).....</b>	<b>56</b>
<b>Интерфейс IApcConnection.....</b>	<b>58</b>
Метод createSession.....	58
Метод createSession2.....	59
<b>Интерфейс IApcSession.....</b>	<b>59</b>
Метод getServer.....	60
Метод close.....	60
Метод set_onDisconnect.....	60

# 1

## Подсистема автоматизации и COM SDK

Раздел содержит информацию о подсистеме автоматизации APACS Bio и COM SDK (комплект для разработки программного обеспечения).

### 1.1 Подсистема автоматизации

Подсистема автоматизации ПК APACS Bio позволяет администратору настроить нестандартное поведение комплекса и автоматизировать реакции системы на часто возникающие ситуации. Подсистема позволяет настраивать как последовательность действий, которые исполняются в комплексе без участия дежурного оператора (в фоновом режиме), так и действия, которые исполняются по решению оператора.

При помощи подсистемы автоматизации могут быть реализованы следующие типичные задачи:

взаимодействие между различными аппаратными системами
уведомление администратора о поступлении определенных сообщений (например, рассылка сообщений по электронной почте)
создание собственного пользовательского интерфейса
передача поступающих сообщений из ПК APACS Bio во внешние системы учета рабочего времени
исполнение в системе каких-либо функций по расписанию

Также подсистема автоматизации позволяет инициировать запись видео на поступление определенных сообщений и связывать записанный видеофрагмент и его сообщение-инициатор (см. Просмотр видео в ПК APACS Bio).

Таким образом, с помощью подсистемы автоматизации администратор может легко настроить комплекс под требования конкретной системы управления и контроля доступом.

#### **Расположение подсистемы автоматизации**

Подсистема автоматизации может быть настроена и запущена в разных частях комплекса:

- на сервере APACS Bio (реализовано в составе [службы «Автоматизация»](#)),
- в рамках разных клиентских приложений APACS Bio, как правило, это приложения «Консоль» и «Дежурный режим» (реализовано в виде модуля **Клиентская автоматизация**),
- на разных рабочих станциях.

На сервер APACS Bio должны быть вынесены безусловные задачи автоматизации, которые должны исполняться в фоновом режиме и не

требуют участия оператора. Ориентированные на пользователя задачи автоматизации требуется настраивать в рамках конкретного приложения.

### **Макрокоманда**

*Макрокоманда* позволяет задавать действие или последовательность действий с объектами или группой объектов, в системе макрокоманды выполняется по решению оператора.

Заданные администратором макрокоманды могут отображаться в меню «Макрокоманды» окна **Основная панель** в приложении «Дежурный режим». В зависимости от ситуации оператор выбирает нужную ему макрокоманду, и команда выполняется в системе.

**Например,** в меню «Макрокоманды» находятся следующие пункты:

- Включить тревогу на складе,
- Поставить офис на охрану,
- Снять офис с охраны,
- Разблокировать все двери в случае пожара.

Макрокоманда может использоваться в качестве составной части при конфигурировании *программ*.

### **Программа**

*Программа* позволяет задать последовательность действий системы на поступающие сообщения. Заданные администратором программы исполняются автоматически и не требуют участия дежурного оператора.

**Обратите внимание:** далее приводятся примеры скриптов. Эти и другие примеры скриптов находятся в папке Samples корневого каталога APACS Bio.

### **Способы конфигурирования подсистемы автоматизации**

Подсистема автоматизации может быть сконфигурирована следующим образом:

- с помощью объектов [службы «Автоматизация»](#),
- с помощью сценариев, написанных на языке VBScript.

Интерфейс APACS Bio позволяет быстро и наглядно настроить программы и макрокоманды, но имеет ограниченный набор действий.

При написании сценариев на языке VBScript администратор может использовать все возможности VBScript: массивы, операторы управления, перечисления свойств объектов, функции работы со строками, обработка ошибок.

VBScript предоставляет доступ к COM-объектам из скрипта, что позволяет использовать возможности других приложений. Например, используя возможности приложения Microsoft Office, можно составлять интерактивные отчеты о сообщениях и объектах системы в MS Word и таблицах MS Excel.

Администратор комплекса может выбрать подходящий ему способ настройки подсистемы автоматизации в зависимости от поставленной задачи.

1.1.1 Конфигурирование подсистемы автоматизации

Подсистема автоматизации конфигурируется в окне **Проводник** с помощью [объектов автоматизации](#).  
Далее мы рассмотрим последовательность конфигурирования автоматизации на сервере и клиенте APACS Bio.

Создание макрокоманды

- Макрокоманда может быть создана следующими способами:
- с помощью интерфейса APACS Bio – создайте в окне **Проводник** объект типа [Простая макрокоманда](#).
  - с помощью языка VBScript:
    - напишите макрокоманду на языке VBScript,
    - создайте объект типа [Макрокоманда VBScript](#) и загрузите в него написанную на VBScript макрокоманду.

Создание программы

- Программа может быть создана следующими способами:
- с помощью интерфейса APACS Bio – создайте в окне **Проводник** объект типа [Простая программа](#).
  - с помощью языка VBScript:
    - напишите программу на языке VBScript,
    - создайте объект типа [Программа VBScript](#) и загрузите в него написанную на VBScript программу.

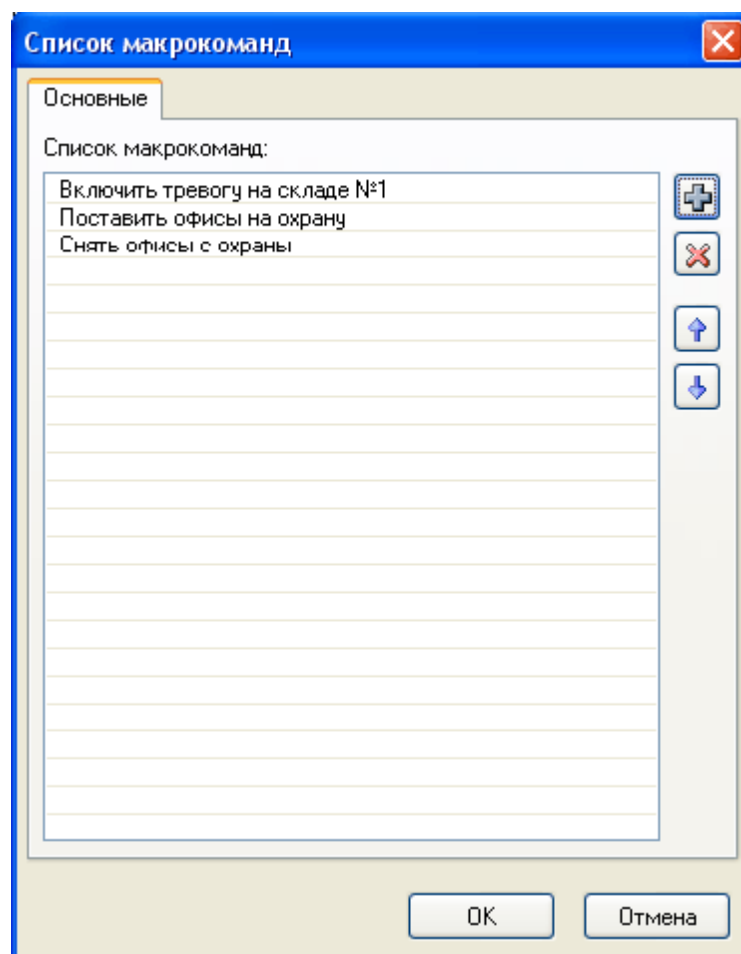


Способы конфигурирования программы и макрокоманды

Конфигурирование списка макрокоманд для приложения

- Для того чтобы в приложении дежурного оператора в меню «Макрокоманды» окна **Основная панель** присутствовал список доступных макрокоманд, выполните следующее:
- Создайте необходимое количество объектов типа [Простая макрокоманда](#) и/или [Макрокоманда VBScript](#).
  - В рамках этого приложения выберите пункт меню «Клиентская автоматизация / Список макрокоманд» окна **Основная панель**.  
Откроется диалоговое окно **Список макрокоманд**, где требуется

сформировать список макрокоманд для данного приложения.



Окно **Список макрокоманд**

- Чтобы подсистема автоматизации работала в рамках этого приложения, в схему приложения включите модуль **Клиентская автоматизация** (объект типа *Схема клиентского приложения* вкладка «**Основные**»).

### **Конфигурирование автоматизации**

Чтобы сконфигурировать автоматизацию, выполните следующее:

- Создайте необходимое количество объектов типа [Простая макрокоманда](#) и/или [Программа VBScript](#).
- Спланируйте выполнение программ в системе с помощью объекта типа [Настройки автоматизации](#).
- Если автоматизация должна выполняться в рамках приложения, выполните следующее:
  - Укажите объект, содержащий описание автоматизации для данного приложения. Для этого пунктом меню «Клиентская автоматизация / Настройки» окна **Основная панель** откройте диалоговое окно **Выбор настроек клиентской автоматизации** и укажите объект.
  - В схему этого приложения включите модуль **Клиентская автоматизация** (объект типа *Схема клиентского приложения* вкладка «**Основные**»).
  - Запустите выполнение автоматизации пунктом меню «Клиентская автоматизация / Перезапустить» окна **Основная панель**. Остановить выполнение автоматизации можно пунктом меню «Клиентская автоматизация / Остановить».





Схема конфигурирования клиентской автоматизации

- Если автоматизация должна выполняться на сервере APACS Bio, выполните следующее:
  - В настройках системного объекта *Настройки сервисов* укажите объект типа [Настройки автоматизации](#), содержащий описание автоматизации, которая должна выполняться на сервере APACS Bio.
  - Чтобы настройки серверной автоматизации вступили в силу, на объекте *Настройки сервисов* выполните команду **Перезапустить автоматизацию**.

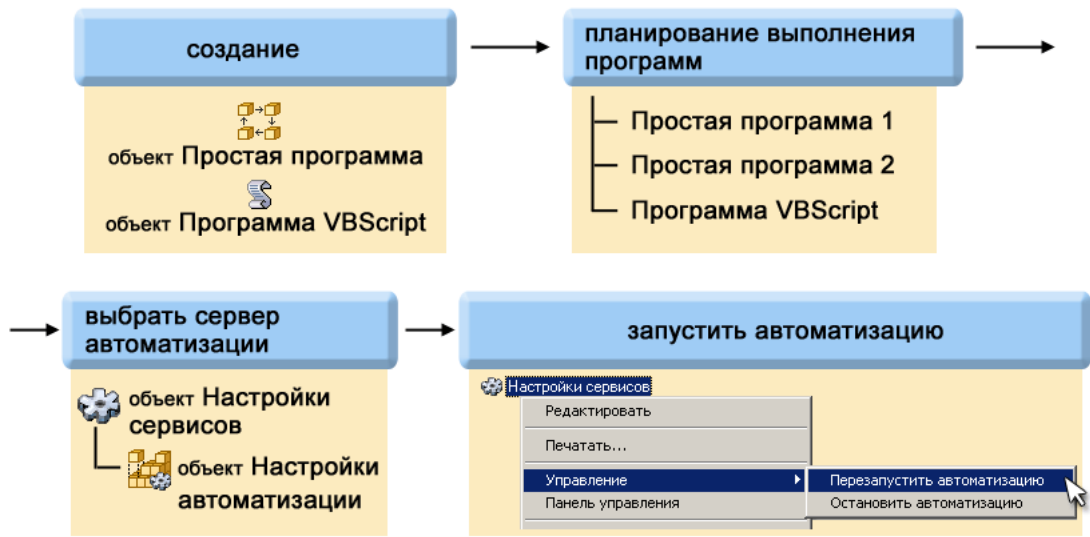


Схема конфигурирования серверной автоматизации

**Обратите внимание:** от клиентской автоматизации сообщения в систему не поступают и не отображаются в модуле «Список сообщений».

1.1.2 Служба «Автоматизация»

Раздел содержит описание объектов службы «Автоматизация». С помощью этих объектов осуществляется конфигурирование [подсистемы автоматизации](#).

1.1.2.1 Настройки автоматизации

**Файл** типа **Настройки автоматизации** позволяет спланировать выполнение программ в системе APACS Bio.

**Настройки**

Все настройки объекта расположены на вкладках

Общие

Основные

Полномочия

Клиентские команды	Описание
<b><i>Свойства файла</i></b>	С помощью этой команды можно открыть окно свойств файла.

1.1.2.2 Макрокоманда VBScript

**Файл** типа **Макрокоманда VBScript** содержит написанный на языке VBScript сценарий управления объектами, который можно запустить на выполнение вручную.

**Настройки**

Все настройки объекта расположены на вкладках

Общие

Импорт/экспорт

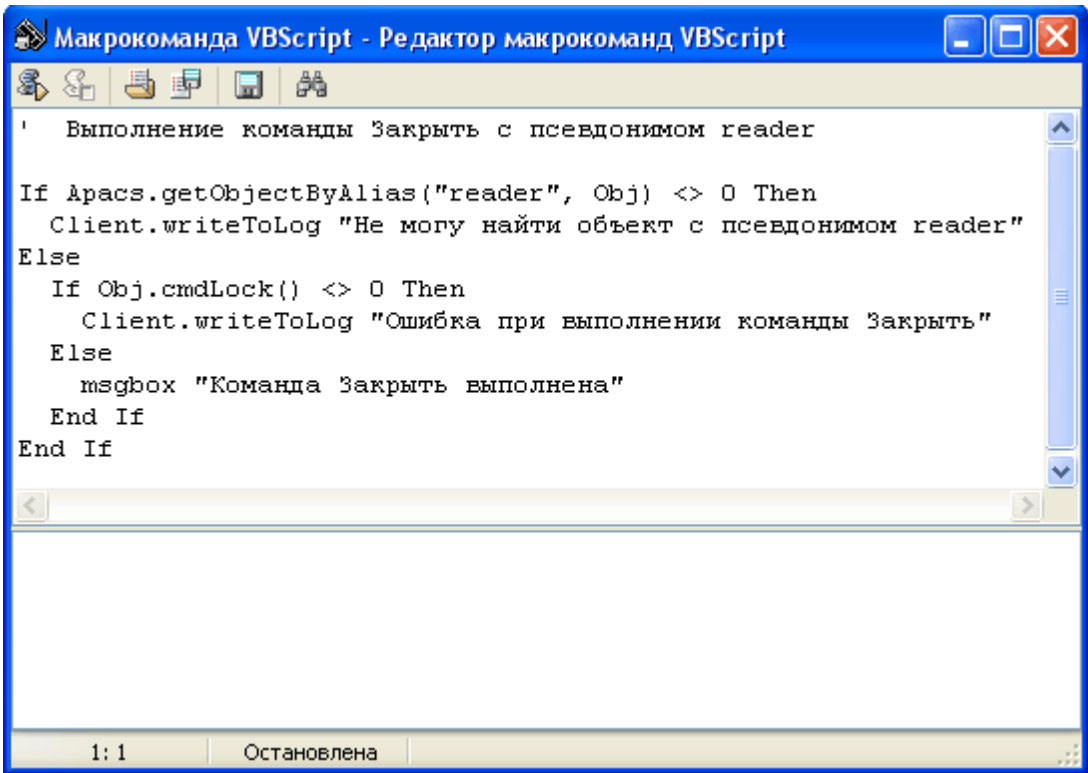
Полномочия

Команды управления	Описание
<b><i>Выполнить</i></b>	Макрокоманда будет выполнена в рамках клиентского приложения (для этого в рамках приложения должна быть настроена клиентская автоматизация).
Клиентские команды	Описание
<b><i>Показать объект на плане</i></b>	При выполнении команды в окне <b><i>Просмотр планов</i></b> откроется план, на котором будет выделен этот объект.
<b><i>Поместить на план статическую иконку</i></b>	Команда позволяет поместить на план статическую иконку объекта.
<b><i>Свойства файла</i></b>	С помощью этой команды можно открыть окно свойств файла.

1.1.2.2.1 Редактор макрокоманд VBScript

Чтобы сконфигурировать объект, выполните на объекте команду ***Редактировать***.

Откроется диалоговое окно ***Редактор макрокоманд VBScript***.



Окно **Редактор макрокоманд VBScript**

- Окно **Редактор макрокоманд VBScript** поделено на две части:
- верхняя часть предназначена для текста макрокоманды, написанной на VBScript,
  - нижняя предназначена для отображения информации об исполнении макрокоманды.
- Внизу окна **Редактор макрокоманд VBScript** располагается строка состояния со следующей информацией:
- позиция курсора,
  - сведения о состоянии макрокоманды (например, *Остановлена*).

Для работы с окном используется панель инструментов.

1.1.2.3 Простая макрокоманда

**Файл** типа **Простая макрокоманда** позволяет задать список объектов и команд, которые должны быть последовательно выполнены на этих объектах.

**Настройки**

- Все настройки объекта расположены на вкладках
- Общие
  - Основные
  - Полномочия

Команды управления	Описание
<b>Выполнить</b>	Макрокоманда будет выполнена в рамках клиентского приложения (для этого в рамках приложения должна быть настроена клиентская автоматизация).

Клиентские команды	Описание
<b>Показать объект на плане</b>	При выполнении команды в окне <b>Просмотр планов</b> откроется план, на котором будет выделен этот объект.
<b>Поместить на план статическую иконку</b>	Команда позволяет поместить на план статическую иконку объекта.
<b>Свойства файла</b>	С помощью этой команды можно открыть окно свойств файла.

1.1.2.4 Программа VBScript

**Файл** типа **Программа VBScript** позволяет задать последовательность действий системы в зависимости от поступающих сообщений, с использованием всех возможностей VBScript.

**Настройки**

- Все настройки объекта расположены на вкладках
- Общие
- Импорт/экспорт
- Полномочия

Клиентские команды	Описание
<b>Свойства файла</b>	С помощью этой команды можно открыть окно свойств файла.

1.1.2.5 Простая программа

**Файл** типа **Простая программа** позволяет с помощью фильтров событий и макрокоманд задать реакции системы на поступающие сообщения.

**Настройки**

- Все настройки объекта расположены на вкладках
- Общие
- Основные
- Полномочия

Клиентские команды	Описание
<b>Свойства файла</b>	С помощью этой команды можно открыть окно свойств файла.

1.1.2.6 Фильтр событий

**Файл** типа **Фильтр событий** позволяет создать условия фильтрации поступающих сообщений, которые будут использоваться в работе простой программы.

**Настройки**

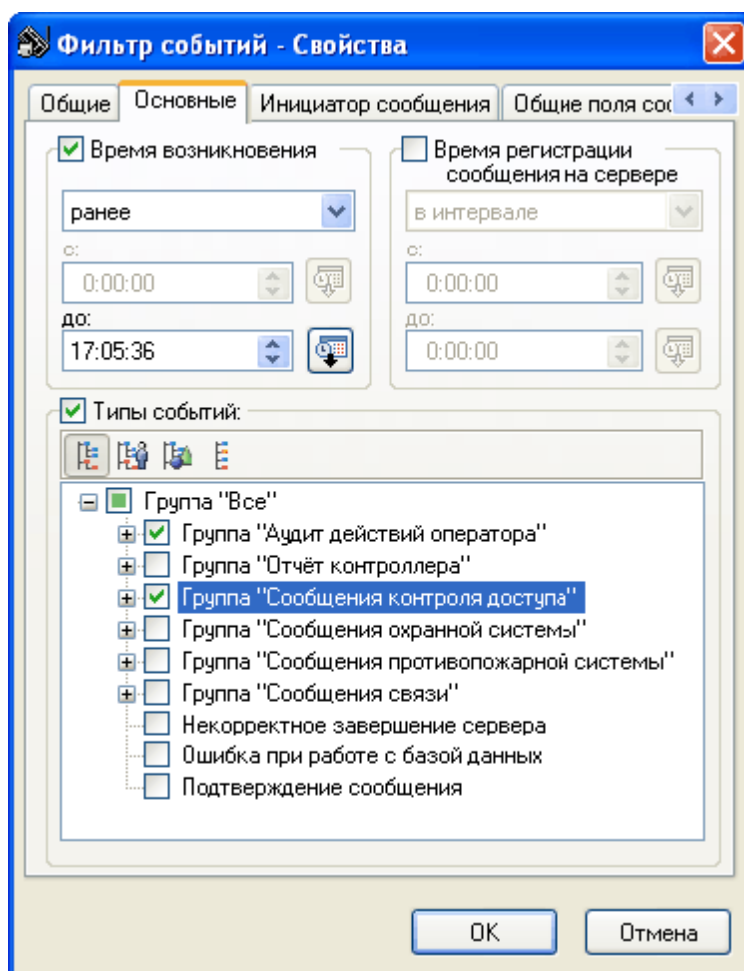
- Все настройки объекта расположены на вкладках
- Общие
- [Основные](#)
- [Инициатор сообщений](#)
- [Общие поля сообщений](#)
- Полномочия

Клиентские команды	Описание
<b>Свойства файла</b>	С помощью этой команды можно открыть окно свойств файла.

1.1.2.6.1 Вкладка «Основные»

На вкладке «**Основные**» окна **Фильтр событий - Свойства** можно задать следующие условия фильтрации сообщений:

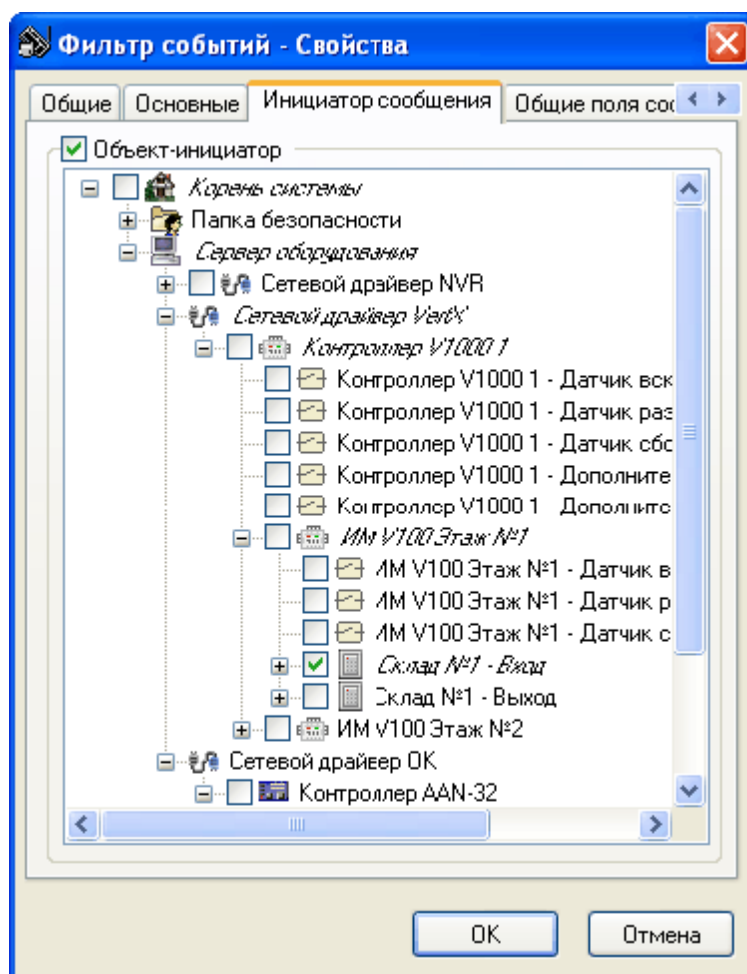
- **Время возникновения** — в этой группе параметров можно задать условие фильтрации по времени возникновения сообщения. В полях **с** и **до** можно указать начало и конец интервала. Также интервал поиска можно ограничить сроком ранее / позднее какого-либо времени. При помощи кнопки **Установить текущее время** можно указать в поле текущее время.
- **Время регистрации сообщения на сервере** — в этой группе параметров можно задать условие фильтрации по времени регистрации сообщения на сервере. В полях **с** и **до** можно указать начало и конец интервала. Также интервал поиска можно ограничить сроком ранее / позднее какого-либо времени. При помощи кнопки **Установить текущее время** можно указать в поле текущее время.
- **Типы событий** — в этом поле укажите нужные типы сообщений.  
Для отображения типов сообщений используются следующие режимы:
  - группировка типов сообщений на логические группы,
  - группировка в зависимости от наличия в сообщении информации о владельце карты,
  - группировка по типам объектов, инициирующих сообщения,
  - список типов сообщений без группировки.Режимы группировки позволяют быстро выбрать необходимые типы сообщений. Задать режим группировки можно при помощи кнопок фильтра.  
Изменение режима не ведет к изменению типов сообщений, выбранных в предыдущем режиме.



Вкладка «**Основные**» окна редактирования свойств файла *Фильтр событий*

#### 1.1.2.6.2 Вкладка «Инициатор сообщения»

На вкладке «**Инициатор сообщения**» окна **Фильтр событий - Свойства** можно выбирать сообщения в зависимости от их объекта-инициатора.



Вкладка «**Инициатор сообщения**» окна редактирования свойств файла *Фильтр событий*

В поле **Объект-инициатор** находится иерархический список объектов, от которых могут поступать сообщения.

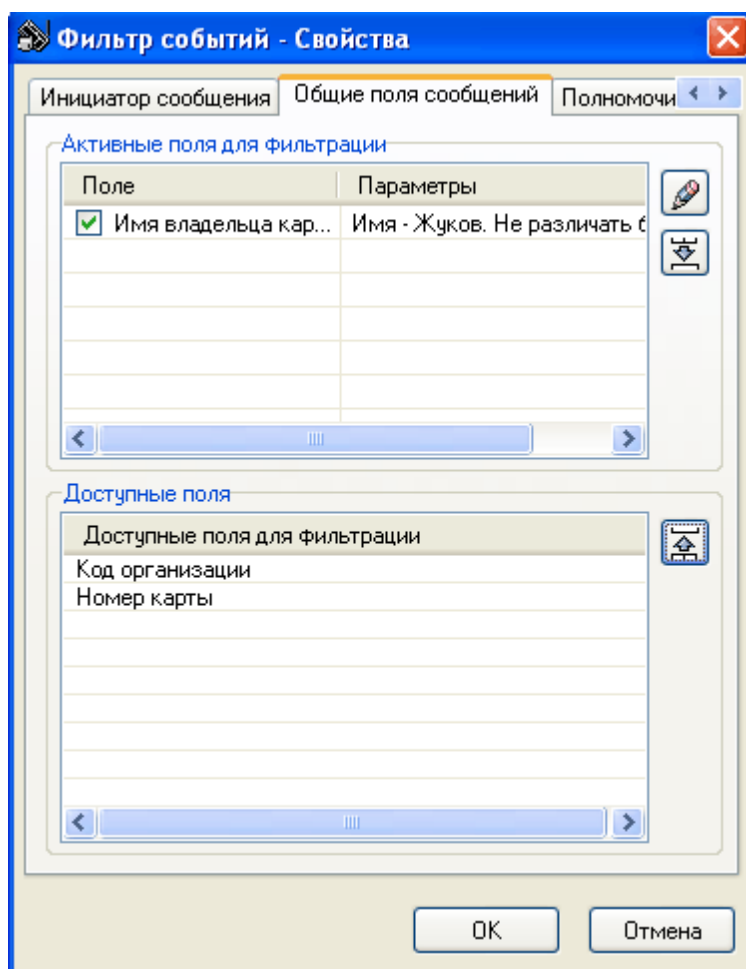
Для выбора объектов удобно использовать контекстное меню, чтобы:

- выделить подэлементы выбранного объекта,
- отменить выделение подэлементов выбранного объекта,
- выделить все объекты, от которых могут поступать сообщения,
- отменить выделение всех выделенных объектов.

#### 1.1.2.6.3 Вкладка «Общие поля сообщений»

На вкладке «**Общие поля сообщений**» окна **Фильтр событий - Свойства** можно задать условия фильтрации по полям сообщений.

В поле **Доступные поля** находятся поля, общие для выбранных ранее типов сообщений. Выберите поле сообщений и перенесите его в поле **Активные поля для фильтрации** кнопкой **Добавить в список активных**.



Вкладка «Общие поля сообщений» окна редактирования свойств файла *Фильтр событий*

Дальнейшие действия зависят от выбранного поля:

- если выбрано поле **Номер карты**, откроется диалоговое окно **Фильтр по номеру карты**, где требуется указать условия выбора сообщений по номеру карты,
- если выбрано поле **Имя владельца карты**, откроется диалоговое окно **Фильтр по имени владельца карты**, где требуется указать условия выбора сообщений по имени владельца карты,
- если выбрано поле **Код организации**, откроется диалоговое окно **Фильтр по коду организации**, где требуется указать условия выбора сообщений по коду организации.

После того как для поля было указано условие фильтрации, поле появится в поле **Активные поля для фильтрации**.

Участвующие в фильтрации поля сообщений отмечаются флажками в поле **Активные поля для фильтрации**. Чтобы отменить фильтрацию по этому полю, снимите флажок.

Чтобы изменить условие фильтрации по этому полю, нажмите кнопку **Изменить**.

Чтобы удалить условие фильтрации по этому полю, нажмите кнопку **Удалить из списка активных**.



## 1.2 Объектная модель, доступная из макрокоманд и программ VBScript

Объектная модель описывает объекты, методы и их параметры, которые могут быть использованы при написании макрокоманд и программ на языке VBScript (далее также скрипты). Методы программ и макрокоманд VBScript носят универсальный характер по отношению к командам, объектам и сообщениям, которые на данный момент поддерживает комплекс.

### Возможности макрокоманд и программ VBScript

Макрокоманды и программы VBScript предоставляют следующие возможности:
получение текущих настроек объектов
редактирование настроек объектов
создание новых объектов
регистрация сообщений от объектов
удаление объектов
управление объектами (выполнение команд)
получение текущих сообщений
получение уведомлений об удалении, изменении, добавлении объектов
получение сообщений из базы данных по фильтру

Для идентификации параметров объектов и сообщений APACS Bio в скрипте используется список свойств следующего вида: имя (идентификатор) – значение.

Параметры могут быть следующих типов:

- **Числовые.** Могут иметь префиксы:
  - **n** — числовые параметры типа Integer (целочисленный, 32 бита, знаковый).  
Например, параметр **Номер объекта** объекта *Корневой элемент системы* имеет идентификатор nSiteID.
  - **b** — числовые параметры типа Byte (8 бит, беззнаковый).  
Например, параметр **Адрес связи через СОМ-порт** объекта *Контроллер AAN-100* имеет идентификатор bCommAddress.
  - **dw** — числовые параметры типа DWord (32 бита, беззнаковый).  
Например, параметр **Максимальное количество карт** объекта *Контроллер AAN-100* имеет идентификатор dwMaxNumCards,
  - **w** — числовые параметры типа Word (16 бит, знаковый).  
Например, параметр **Длительность импульса** объекта *Дополнительное реле считывателя* имеет идентификатор wPulseTime.
- **Строковые,** имеют префикс **str** (например, общее для всех сообщений свойство **Тип сообщения** имеет идентификатор strEventTypeID),
- **Бинарные массивы,** имеют префикс **blob** (например, параметр **Термы со значением ИСТИНА** объекта *Внутренняя переменная* имеет идентификатор blobTermsValue),

**Обратите внимание:** массивы, используемые в системе APACS Bio, не доступны из скриптов. Для работы с массивами необходимо использовать метод `Script.toVBAArray( cppArray)` для преобразования массивов из формата APACS Bio в формат, типичный для VBScript.

- **Дата и время**, имеют префикс **dt** (например, общий для всех объектов параметр **Дата создания** имеет идентификатор `dtCreateTime`).
- **Ссылка на объект конфигурации**, имеют префикс **SysAddr** (например, параметр **Первый формат** объекта *Список форматов карт* имеет идентификатор `SysAddrCardFormats1`).  
При этом значения могут являться:
  - ссылкой на объект системы (например, в выпадающем списке **Первый формат** объекта *Список форматов карт* выбрано *Формат карт №1*),
  - NULL — в том случае, если данный параметр объекта не используется (в окне редактирования свойств объекта для параметра указано *не используется*, например, в поле **Первый формат** объекта *Список форматов карт* выбрано *не используется*).

**Обратите внимание:** идентификаторы объектов и сообщений, их свойств и команд, которые могут быть к ним применены, можно посмотреть:

- в окне **Просмотр метаданных**,
- на странице электронной справки, вызванной к объекту,
- идентификатор конкретного экземпляра объекта можно посмотреть на вкладке **Общие**, в поле **Системный адрес для SDK**,
- идентификатор конкретного экземпляра сообщения можно посмотреть на вкладке **«Сообщение»**, в поле **Идентификатор для SDK**.

### 1.2.1 Получение доступа к объектам и сообщениям из скрипта

#### **Получение доступа к серверным объектам APACS Bio из скрипта**

Для получения доступа к объектам APACS Bio из скрипта (сценария автоматизации, написанного на языке VBScript) можно воспользоваться следующими способами:

- использовать метод `getObjectByAlias`. Для использования этого метода в свойствах объекта обязательно требуется указать *псевдоним* – уникальный текстовый идентификатор объекта (окно редактирования свойств объекта, вкладка **«Общие»** поле **Псевдоним**). Псевдоним требуется задавать буквами латинского алфавита.
- вызвать метод `getRootObject`, который позволяет получить ссылку на *Корневой элемент системы*, и далее использовать методы:
  - `getChildrenObjsByTypes` — позволяет получить список дочерних объектов указанного типа, существующих на текущий момент в конфигурации системы у данного объекта,
  - `getChildrenObjs` — позволяет получить список всех дочерних объектов данного объекта, существующих на данный момент в конфигурации системы.
- использовать метод `getObjectsByFilter`, который позволяет получить объекты определенного типа в соответствии с поставленным условием.

#### **Получение доступа к сообщениям APACS Bio**

Чтобы разработать скрипт, в который поступают сообщения APACS Bio, используйте следующее:

- для получения текущих сообщений переопределите тело функции ***Apacs\_Event(PropEvt)***,
- для получения доступа к сообщениям из базы данных используйте метод `getEvents`.

**Получение уведомлений**

Для получения уведомлений об изменении объектов APACS Bio используйте следующие методы:

- `set_onNotifyAdd` — метод позволяет установить обработчик, который будет принимать уведомления о добавлении объектов,
- `set_onNotifyChange` — метод позволяет установить обработчик, который будет принимать уведомления об изменении настроек объектов,
- `set_onNotifyDelete` — метод позволяет установить обработчик, который будет принимать уведомления об удалении объектов.

Для получения уведомлений о сообщениях используйте метод `set_onEvent`.

**1.2.2 Предопределенные функции в программах VBScript**

В программах VBScript предопределены следующие функции:

Функция	Описание
<a href="#"><u><i>Apacs_Event(PropEvt)</i></u></a>	Основная функция программы VBScript. После того как функция была определена, она будет вызываться каждый раз когда в системе APACS Bio регистрируется новое сообщение. В теле функции можно задать последовательность действий системы и условия, при которых эти действия должны выполняться.
<a href="#"><u><i>Script_Load()</i></u></a>	Функция позволяет получать оповещения о загрузке программы.
<a href="#"><u><i>Script_Close()</i></u></a>	Функция позволяет получать уведомление о том, что скрипт будет сейчас выгружен.

**1.2.2.1 Функция Apacs\_Event(PropEvt)**

***Apacs\_Event(PropEvt)***,

где **PropEvt** — список текущих свойств сообщения.  
Основная функция программы VBScript. После того как функция была определена, она будет вызываться каждый раз когда в системе APACS Bio регистрируется новое сообщение.  
В теле функции можно задать последовательность действий системы и условия, при которых эти действия должны выполняться.

**Пример обработчика на поступление сообщения**

```
Sub Apacs_Event( Obj)
    MsgBox( "          " + Obj.strEventTypeID + "          " + Obj.strInitObjName)
End Sub
```

**Пример реакции системы на сообщение** *Доступ запрещен, необходимо*



	выводить отладочную информацию в окно утилиты «Просмотр логов». Объект поддерживает интерфейс <a href="#">IApcLogWrap</a> .
<i>Client</i>	Методы объекта Client предоставляют доступ к локальным ресурсам компьютера, на котором запущено клиентское приложение APACS Bio. Объект реализован только на клиенте APACS Bio, и не поддерживается на сервере APACS Bio. Объект поддерживает интерфейс <a href="#">IApcClientWrap</a> .
<i>Script</i>	Поддерживает интерфейс <a href="#">IApcScriptWrap</a> .
<i>VideoLinker</i>	Поддерживает интерфейс <a href="#">IApcVideoLinkerWrap</a> , имеет ряд вспомогательных функций, предоставляет интерфейс к связыванию видеофрагментов.

Вспомогательные	
Объект	Описание
Объекты системы	Поддерживают интерфейс <a href="#">IApcObjectWrap</a> .
класс <i>TApcEQUALObjFilter</i>	Экземпляры этого класса позволяют задать фильтр на эквивалентность, реализуют интерфейс <a href="#">IApcEQUALObjFilter</a> .
класс <i>TApcANDObjFilter</i>	Экземпляры этого класса позволяют объединить фильтры по логическому И, реализуют интерфейс <a href="#">IApcANDObjFilter</a> .

Объекты	Клиентская автоматизация	Серверная автоматизация
Аpacs	✓	✓
Log	✓	✓
Client	✓	✗
Script	✓	✓
VideoLinker	не рекомендуется	✓

Таблица показывает, где могут быть использованы предопределенные объекты VBScript

1.2.3.1 Интерфейс IApcServerWrap

Интерфейс сервера APACS Bio. Предоставляет следующие методы:

Метод	Описание
<a href="#">getObjectByAlias</a>	Метод предназначен для получения ссылки на объект по его псевдониму.
<a href="#">getRootObject</a>	Метод предназначен для получения ссылки на <i>Корневой элемент системы</i> .
<a href="#">getObjectByUID</a>	Метод предназначен для получения ссылки на объект по его строковому идентификатору в системе.
<a href="#">getObjectsByFilter</a>	Метод позволяет получить объекты указанного типа в соответствии с условием.
<a href="#">getEvents</a>	Метод позволяет получать сообщения из базы данных.
<a href="#">set_onEvent</a>	Метод позволяет установить обработчик, который будет принимать сообщения, происходящие в системе APACS Bio.
<a href="#">set_onNotifyAdd</a>	Метод позволяет установить обработчик, который будет принимать уведомления о добавлении объектов в системе APACS Bio.
<a href="#">set_onNotifyChange</a>	Метод позволяет установить обработчик, который будет принимать уведомления об изменении настроек объектов в системе APACS Bio.
<a href="#">set_onNotifyDelete</a>	Метод позволяет установить обработчик, который будет принимать уведомления об удалении объектов в системе APACS Bio.

1.2.3.1.1 Метод getObjectByAlias

Метод предназначен для получения ссылки на объект по его псевдониму.

VBScript

```
getObjectByAlias( strAlias, Obj)
где strAlias — псевдоним объекта,
Obj — возвращаемое значение, ссылка на объект системы.
```

Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

Возможные причины невыполнения

- нет связи с сервером,
- объекта с таким псевдонимом не существует в системе,

- нет прав на просмотр этого объекта.

### Пример доступа к объекту системы из скрипта

```

,
sample
If Apacs.GetObjectByAlias( "sample", obj) <> 0 Then
    Client.writeToLog "sample"
Else
    If obj.GetCurrentSettings( prop) <> 0 Then
        Client.writeToLog " "
    Else
        MsgBox " : " & prop.strName
    End If
End If
End If

```

### C++

```

HRESULT getObjectByAlias( BSTR astrObjAlias/*[in]*/,
                          LPDISPATCH* appObject/*[out]*/,
                          int* apnResult/*[out,retval]*/)

```

### Параметры

[in] astrObjAlias — псевдоним объекта,  
 [out] appObject — указатель на объект, поддерживающий интерфейс [IApcObjectWrap](#),  
 [out] apnResult — успешное выполнение — 0, иначе — код ошибки.

### Возвращаемое значение

Возвращает значение HRESULT.

#### 1.2.3.1.2 Метод getRootObject

Метод предназначен для получения ссылки на *Корневой элемент системы*.

### VBScript

```

getRootObject( RootObj)
где RootObj — возвращаемое значение, ссылка на Корневой элемент системы.

```

### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

### Возможные причины невыполнения

- нет связи с сервером.

### Пример изменения имени корня системы

```

If Apacs.getRootObject( objRoot) <> 0 Then
    Client.writeToLog " "
Else
    If objRoot.GetCurrentSettings( prop) <> 0 Then
        Client.writeToLog " "
    Else
        prop.strName = " "
        If objRoot.applySettings( prop) <> 0 Then
            Client.writetoLog " "
        Else
            MsgBox " "
        End If
    End If
End If
End If

```

**C++**

```
HRESULT getObject( LPDISPATCH* appObject/*[out]*/,
                  int* apnResult/*[out,retval]*/)
```

**Параметры**

[out] appObject — указатель на объект, поддерживающий интерфейс [IApcObjectWrap](#),

[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

**1.2.3.1.3 Метод getObjectByUID**

Метод предназначен для получения ссылки на объект по его строковому идентификатору в системе.

**VBScript**

```
getObjectByUID( strUID, Obj)
```

где strUID — строковое представление идентификатора объекта,  
Obj — возвращаемое значение, ссылка на объект системы.

Формат строкового представления идентификатора объекта (strUID) имеет следующий вид: SA XXXX.YYYYYYYY, где XXXX — первая часть системного адреса в шестнадцатеричном формате, а YYYYYYYY — вторая часть системного адреса в шестнадцатеричном формате.

**Например**, строковое представление идентификатора объекта может быть таким: SA 0000.0000000A

**Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

**Возможные причины невыполнения**

- нет связи с сервером,
- объекта с таким strUID не существует в системе.

**Пример**

```
' UID SA 0000.00000001
If Apacs.getObjectByUID( "SA 0000.00000001", obj) <> 0 Then
    Client.writeToLog " UID SA 0000.00000001"
Else
    If obj.getCurrentSettings( prop) <> 0 Then
        Client.writeToLog " "
    Else
        msgbox " : " & prop.strName
    End If
End If
```

**C++**

```
HRESULT getObjectByUID( BSTR astrUID/*[in]*/,
                       LPDISPATCH* appObject/*[out]*/,
                       int* apnResult/*[out,retval]*/)
```

**Параметры**

[in] astrUID — идентификатор объекта,

[out] appObject — указатель на объект, поддерживающий интерфейс [IApcObjectWrap](#),



[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

### Возвращаемое значение

Возвращает значение HRESULT.

#### 1.2.3.1.4 Метод getObjectByFilter

Метод позволяет получить объекты указанного типа в соответствии с условием apFilter. В процессе выполнения метод формирует SQL-запрос.

#### VBScript

getObjectByFilter ( astrObjType, apFilter, appObjs)

где astrObjType — тип объектов для фильтрации,

apFilter — условие фильтрации, в качестве фильтра могут выступать объекты, поддерживающие интерфейсы [IApcEQUALObjFilter](#) и [IApcANDObjFilter](#),

если в качестве значения параметра apFilter задано nothing, то возвращаться будут все записи без применения фильтра,

appObjs — возвращаемое значение, список объектов, удовлетворяющих фильтру.

### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

### Возможные причины невыполнения

- нет связи с сервером.

### Пример получения владельцев карт по условию фильтра

```

'
'
set objFirstNameCondition = CreateObject( "ApcSrvSDK.TApcEQUALObjFilter")
objFirstNameCondition.strName = "strFirstName"
objFirstNameCondition.Value = "          "
'

set objMiddleNameCondition = CreateObject( "ApcSrvSDK.TApcEQUALObjFilter")
objMiddleNameCondition.strName = "strMiddleName"
objMiddleNameCondition.Value = "          "
'

set objANDFilter = CreateObject( "ApcSrvSDK.TApcANDObjFilter")
objANDFilter.addCondition objFirstNameCondition
objANDFilter.addCondition objMiddleNameCondition
'

If Apacs.getObjectByFilter( "TApcCardHolder", objANDFilter, listObjs) <> 0 Then
    Client.writeToLog "          "
Else
    first = true
    strRes = ""
    For i = 0 To UBound ( listObjs)
        If listObjs( i) .getCurrentSettings( prop) = 0 Then
            If first Then
                strRes = prop.strLastName
                first = false
            Else
                strRes = strRes + vbNewLine + prop.strLastName
            End If
        End If
    Next
    MsgBox "          : " + vbNewLine + strRes
End If

```

**Пример получения владельцев карт по фильтру без условия**

```

if Apacs.getObjectsByFilter( "TApcCardHolder", nothing, listH) <> 0 then
    msgError "getObjectsByFilter fail"
    exit function
End if

```

**C++**

```

HRESULT getObjectsByFilter( BSTR astrObjType/*[in]*/,
                           LPDISPATCH apFilter/*[in]*/,
                           LPSAFEARRAY* appObjs/*[out]*/,
                           int* apnResult/*[out,retval]*/)

```

**Параметры**

[in] astrObjType — тип объектов для фильтрации,  
[in] apFilter — условие фильтрации, в качестве фильтра может выступать объекты, поддерживающие интерфейсы [IApcEQUALObjFilter](#) и [IApcANDObjFilter](#),  
[out] appObjs — возвращаемое значение, список объектов, удовлетворяющих фильтру,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

**1.2.3.1.5 Метод getEvents**

Метод позволяет получать сообщения из базы данных.

**VBScript**

```
getEvents ( arrEventTypes, dtFrom, dtTo, arrEvents )
```

где arrEventTypes — список типов сообщений,

adtFrom — начиная с этого момента времени будем получать сообщения,

adtTo — момент окончания получения сообщений,

arrEvents — возвращаемое значение, список сообщений.

**Обратите внимание:** временем прихода сообщения считается время его возникновения, а не время регистрации на сервере.

**Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

**Пример получения общего количества сообщений типа *Доступ разрешен за день***

```

If Apacs.getEvents( Array( "TApcCardHolderAccess__Granted" ) , DateAdd( "d", -1, Now( ) ) , Now( ) ,
    evtProps) <> 0 Then
    Client.writeToLog "
Else
    msgbox " : " & ( UBound( evtProps) + 1)
End If

```

**C++**

```

HRESULT getEvents( LPSAFEARRAY aEventTypes/*[in]*/,
                  DATE adtFrom/*[in]*/,
                  DATE adtTo/*[in]*/, LPSAFEARRAY* appEvents/*[out]*/,
                  int* apnResult/*[out,retval]*/)

```

### Параметры

[in] aEventTypes — массив строк типов сообщений,  
[in] adtFrom — начиная с этого момента времени будем получать сообщения,  
[in] adtTo — момент окончания получения сообщений,  
[out] appEvents — список сообщений,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

### Возвращаемое значение

Возвращает значение HRESULT.

#### 1.2.3.1.6 Метод set\_onEvent

Метод позволяет установить обработчик, который будет принимать сообщения, происходящие в системе APACS Bio. При возникновении сообщения в системе для объекта, переданного в функцию, будет вызван метод Invoke с DispID=0 и с параметром « ». Метод рекомендуется использовать в рамках COM SDK.

#### VBScript

```
set_onEvent ( subRef)
```

где subRef — указатель на функцию, которая будет вызвана при поступлении сообщения в системе.

### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

### Пример установки обработчика, принимающего сообщения системы

```
Apacs.set_onEvent getref( "Apacs__Event")
```

```
sub Apacs__Event( Evt)
```

```
Client.writeToLog " " & Evt.SysAddrEventID & " " &
```

```
Evt.strEventTypeID
```

```
end sub
```

**Обратите внимание:** если Вы установили обработчик событий, перед завершением использования интерфейса IApcServerWrap, чтобы избежать за цикливания объектных ссылок, требуется установить обработчик в nothing.

#### C++

```
HRESULT set_onEvent( LPDISPATCH apEventHandler/*[in]*/) )
```

### Параметры

[in] apEventHandler — обработчик сообщений.

### Возвращаемое значение

Возвращает значение HRESULT.

**Обратите внимание:** если Вы установили обработчик событий, перед завершением использования интерфейса IApcServerWrap, чтобы избежать за цикливания объектных ссылок, требуется установить обработчик в NULL.

### 1.2.3.1.7 Метод set\_onNotifyAdd

Метод позволяет установить обработчик, который будет принимать уведомления о добавлении объектов в системе APACS Bio. При добавлении объекта в систему для объекта, переданного в функцию, будет вызван метод Invoke с DispID=0 и с параметром « ».

#### VBScript

```
set_onNotifyAdd ( subRef)
```

где subRef — указатель на функцию, которая будет вызвана при добавлении объекта в системе.

#### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

#### Пример установки обработчика, принимающего уведомления о добавлении объектов в системе

```
Apacs.set_onNotifyAdd getref( "APACS Bio__NotifyAdd")
```

```
Sub APACS Bio__NotifyAdd ( Obj)
```

```
    Obj.GetUID strUID
```

```
    Obj.GetApacsType strType
```

```
    Client.writeToLog "          : " + strUID + " ,          : " + strType
```

```
End Sub
```

**Обратите внимание:** если Вы установили обработчик событий, перед завершением использования интерфейса IApcServerWrap, чтобы избежать заикливания объектных ссылок, требуется установить обработчик в nothing.

#### C++

```
HRESULT set_onNotifyAdd ( LPDISPATCH apEventHandler/*[in]*/) 
```

#### Параметры

[in] apEventHandler — обработчик уведомлений о добавлении объектов.

#### Возвращаемое значение

Возвращает значение HRESULT.

**Обратите внимание:** если Вы установили обработчик уведомления о добавлении объектов, перед завершением использования интерфейса IApcServerWrap, чтобы избежать заикливания объектных ссылок, требуется установить обработчик в NULL.

### 1.2.3.1.8 Метод set\_onNotifyChange

Метод позволяет установить обработчик, который будет принимать уведомления об изменении настроек объектов в системе APACS Bio. При изменении настроек объекта в системе для переданного в функцию объекта будет вызван метод Invoke с DispID=0, с первым параметром « » и со вторым параметром « ».

#### VBScript

```
set_onNotifyChange ( subRef, changedProp)
```

где subRef — указатель на функцию, которая будет вызвана при изменении настроек объектов в системе,

changedProp — список изменившихся настроек объекта.

### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

### Пример установки обработчика, принимающего уведомления об изменении настроек объектов в системе

```
Apacs.set__onNotifyChange getref( "APACS Bio__NotifyChange")
Sub APACS Bio__NotifyChange( Obj, ChangedProp)
    Obj.GetUID strUID
    Obj.GetApacsType strType
    Client.writeToLog strType
End Sub
```

**Обратите внимание:** если Вы установили обработчик событий, перед завершением использования интерфейса IApcServerWrap, чтобы избежать за цикливания объектных ссылок, требуется установить обработчик в nothing.

### C++

```
HRESULT set__onNotifyChange ( LPDISPATCH apEventHandler/*[in]*/) 
```

### Параметры

[in] apEventHandler — обработчик уведомлений об изменении настроек объектов.

### Возвращаемое значение

Возвращает значение HRESULT.

**Обратите внимание:** если Вы установили обработчик уведомления об изменении настроек объектов, перед завершением использования интерфейса IApcServerWrap, чтобы избежать за цикливания объектных ссылок, требуется установить обработчик в NULL.

#### 1.2.3.1.9 Метод set\_onNotifyDelete

Метод позволяет установить обработчик, который будет принимать уведомления об удалении объектов в системе APACS Bio. При удалении объекта в системе для переданного в функцию объекта будет вызван метод Invoke с DispID=0 и с параметром « ».

**Обратите внимание:** так как объект уже будет удален в системе, для этого объекта будет успешно выполнена только функция getUID.

### VBScript

```
set_onNotifyDelete ( subRef)
```

где subRef — указатель на функцию, которая будет вызвана при удалении объекта в системе.

### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

### Пример установки обработчика, принимающего уведомления об удалении объектов в системе

```
Apacs.set__onNotifyDelete getref( "APACS Bio__NotifyDelete")
```

```
Sub APACS Bio__NotifyDelete( Obj)
  Obj.getUID strUID
  Obj.getApacsType strType
  Client.writeToLog "          : " + strUID + " ,          : " + strType
End Sub
```

**Обратите внимание:** если Вы установили обработчик событий, перед завершением использования интерфейса IApcServerWrap, чтобы избежать зацикливания объектных ссылок, требуется установить обработчик в nothing.

C++

```
HRESULT set_onNotifyDelete ( LPDISPATCH apEventHandler/*[in]*/) 
```

**Параметры**  
[in] apEventHandler — обработчик уведомлений об удалении объекта.

**Возвращаемое значение**  
Возвращает значение HRESULT.

**Обратите внимание:** если Вы установили обработчик уведомления об удалении объектов, перед завершением использования интерфейса IApcServerWrap, чтобы избежать зацикливания объектных ссылок, требуется установить обработчик в NULL.

1.2.3.2 Интерфейс IApcLogWrap

Интерфейс объекта *Log*. Предоставляет методы вывода логов. Все логи выводятся в подсистему вывода отладочной информации.

Метод	Описание
<a href="#">logBAD</a>	Метод выводит в логи тревожную информацию: выделенную красным цветом.
<a href="#">logNORM</a>	Метод выводит в логи информацию с умеренным уровнем тревожности: выделенную желтым цветом.
<a href="#">logTRK</a>	Метод выводит в логи не тревожную информацию: выделенную серым цветом.

1.2.3.2.1 Метод logBAD

Метод позволяет отправлять сведения в подсистему вывода отладочной информации.

VBScript

```
logB D( strLog)
где strLog — строка с информацией.
```

**Пример вывода строки в окно утилиты «Просмотр логов»**  
Log.logBAD( " ")

1.2.3.2.2 Метод logNORM

Метод выводит в логи информацию с умеренным уровнем тревожности: выделенную желтым цветом.

VBScript

logNORM( strLog)  
где strLog — строка с информацией.

**Пример вывода строки в окно утилиты «Просмотр логов»**  
Log.logNORM( " " )

1.2.3.2.3 Метод logTRK

Метод выводит в логи не тревожную информацию: выделенную серым цветом.

VBScript

logTRK( strLog)  
где strLog — строка с информацией.

**Пример вывода строки в окно утилиты «Просмотр логов»**  
Log.logTRK( " " )

1.2.3.3 Интерфейс IApcClientWrap

Интерфейс объекта *Client*. Доступен только на клиенте APACS Bio. Предоставляет следующие методы:

Метод	Описание
<a href="#">playSound</a>	Метод предназначен для проигрывания звукового файла.
<a href="#">showVideo</a>	Метод позволяет начать показ видеоизображения.
<a href="#">hideVideo</a>	Метод позволяет завершить показ видеоизображения.
<a href="#">writeToLog</a>	Метод предназначен для вывода строки в окно <b>Журнал работы</b> .

1.2.3.3.1 Метод playSound

Метод предназначен для проигрывания звукового файла. Поддерживаются файлы в формате \*.wav.

VBScript

playSound( strFile)  
где strFile — путь к внешнему файлу.

**Обратите внимание:** путь к внешнему файлу должен быть указан относительно того компьютера, на котором будет проигрываться файл.

**Пример проигрывания файла**  
, C:\Windows\Media\Tada.wav  
Client.playSound "C:\Windows\Media\Tada.wav"

1.2.3.3.2 Метод showVideo

Метод позволяет открыть окно с изображение от указанного видеоисточника.

VBScript

showVideo( objCam, dwLeft, dwTop, dwWidth, dwHeight)

где objCam — объект системы, содержащий информацию о камере,  
dwLeft — положение окна с видеоизображением, отступ от левого края монитора (пиксели),  
dwTop — окна с видеоизображением, отступ от верхнего края монитора (пиксели),  
dwWidth — ширина окна с видеоизображением (пиксели),  
dwHeight — высота окна с видеоизображением (пиксели).

Пример показа видео с камеры с псевдонимом cam

```
If Apacs.GetObjectByAlias( "cam", objCam) <> 0 Then
  Client.writeToLog "cam"
Else
  Client.showVideo objCam, 50, 50, 640, 480
End If
```

1.2.3.3.3 Метод hideVideo

Метод позволяет закрыть окно с видеоизображением.

VBScript

hideVideo( objCam)

где objCam — объект системы, содержащий информацию о камере.

Пример завершения показа видео с камеры с псевдонимом cam

```
If Apacs.GetObjectByAlias( "cam", objCam) <> 0 Then
  Client.writeToLog "cam"
Else
  Client.hideVideo objCam
End If
```

1.2.3.3.4 Метод writeToLog

Метод предназначен для вывода строки в окно **Журнал работы**.

VBScript

writeToLog( strLog)

где strLog — строка с информацией.

Пример вывод строки в окно Журнал работы:

```
Client.writeToLog( " !")
```

1.2.3.4 Интерфейс IApcScriptWrap

Интерфейс объекта *Script*. Предоставляет следующие методы:

Метод	Описание
<a href="#">sleep</a>	Метод позволяет приостанавливать выполнение скрипта на указанное время.
Работа с таймерами	



<a href="#">setTimeout</a>	Метод предназначен для однократного вызова функции через указанное время.
<a href="#">clearTimeout</a>	Метод предназначен для сброса таймера, созданного ранее при помощи метода <b>setTimeout</b> .
<a href="#">setInterval</a>	Метод предназначен для многократного вызова функции через указанный интервал времени.
<a href="#">clearInterval</a>	Метод предназначен для сброса таймера, созданного ранее при помощи метода <b>setInterval</b> .
Работа с массивами	
<a href="#">toVBAArray</a>	Метод предназначен для преобразования массивов, используемых в системе APACS Bio, в формат массивов, принятых в VBScript.
<a href="#">toCPPArray</a>	Метод предназначен для преобразования массивов, используемых в VBScript, в формат массивов, принятых в системе APACS Bio.
<a href="#">loadCPPArray</a>	Метод предназначен для загрузки в переменную APACS Bio массива из указанного файла.
<a href="#">saveCPPArray</a>	Метод предназначен для сохранения в указанный файл массива в формате APACS Bio.

1.2.3.4.1 Метод sleep

Метод позволяет приостанавливать выполнение скрипта на указанное время.

**Обратите внимание:** избегайте длительной приостановки выполнения скрипта, так как сообщения, поступившее за это время, будут не обработаны подсистемой автоматизации.

**VBScript**

```
sleep( dwMillesecond)
где dwMillesecond — время в миллисекундах.
```

Пример

```
Client.writeToLog "      : " + CStr( Now( ) )
Script.sleep( 1000)
Client.writeToLog "      : " + CStr( Now( ) )
```

#### 1.2.3.4.2 Метод setTimeout

Метод предназначен для однократного вызова функции через указанное время.

##### VBScript

setTimeout( subRef, dwMillesecond)

где subRef — указатель на функцию, которая будет однократно вызвана по прошествии указанного времени,

dwMillesecond — время в миллисекундах.

##### Возвращаемое значение

В случае успешного выполнения метод возвращает идентификатор, который можно использовать для отмены вызова функции при помощи метода [clearTimeout](#).

##### Пример работы с методами setTimeout и clearTimeout

```
Script.setTimeout getref( "timer1" ) , 1000
dim id2
id2 = Script.setTimeout( getref( "timer2" ) , 1000)
Script.clearTimeout id2
Sub timer1 ( )
    msgbox "                1"
End Sub
Sub timer2 ( )
    msgbox "                2"
End Sub
```

#### 1.2.3.4.3 Метод clearTimeout

Метод предназначен для сброса таймера, созданного ранее при помощи метода [setTimeout](#).

##### VBScript

clearTimeout( id)

где id — идентификатор, возвращенный при выполнении метода setTimeout.

##### Пример работы с методами setTimeout и clearTimeout

```
Script.setTimeout getref( "timer1" ) , 1000
dim id2
id2 = Script.setTimeout( getref( "timer2" ) , 1000)
Script.clearTimeout id2
Sub timer1 ( )
    msgbox "                1"
End Sub
Sub timer2 ( )
    msgbox "                2"
End Sub
```

#### 1.2.3.4.4 Метод setInterval

Метод предназначен для многократного вызова функции через указанный интервал времени.

##### VBScript

setInterval( subRef, dwMillesecond)

где subRef — указатель на функцию, которая будет многократно вызвана через указанный интервал времени,  
dwMillesecond — время в миллисекундах.

### Возвращаемое значение

В случае успешного выполнения метод возвращает идентификатор, который можно использовать для отмены вызова функции при помощи метода [clearInterval](#).

### Пример работы с методами setInterval и clearInterval

```
dim Count
Count = 0
dim id
id = Script.setInterval( getref( "timer") , 1000)
Sub timer( )
    If Count = 0 Then
        MsgBox "          "
    End If
    If Count = 1 Then
        MsgBox "          "
        Script.clearInterval id
    End If
    If Count > 1 Then
        MsgBox "          "
    End If
    Count = Count + 1
End Sub
```

#### 1.2.3.4.5 Метод clearInterval

Метод предназначен для сброса таймера, созданного ранее при помощи метода [setInterval](#).

#### VBScript

clearInterval( id)

где id — идентификатор, возвращенный при выполнении метода setInterval.

### Пример работы с методами setInterval и clearInterval

```
dim Count
Count = 0
dim id
id = Script.setInterval( getref( "timer") , 1000)
Sub timer( )
    If Count = 0 Then
        MsgBox "          "
    End If
    If Count = 1 Then
        MsgBox "          "
        Script.clearInterval id
    End If
    If Count > 1 Then
        MsgBox "          "
    End If
    Count = Count + 1
End Sub
```

1.2.3.4.6 Метод toVBAArray

Метод предназначен для преобразования массивов, используемых в системе APACS Bio, в формат массивов, принятых в VBScript.

**VBScript**

toVBAArray( cppArray)  
где cppArray — массив в формате, принятом в системе APACS Bio,

**Возвращаемое значение**

В случае успешного выполнения метод возвращает массив в формате, принятом в VBScript.

**Пример**

```
'
iv

If Apacs.GetObjectByAlias( "iv", iv) <> 0 Then
    Client.writeToLog "iv"
Else
    If iv.GetCurrentSettings( prop) <> 0 Then
        Client.writeToLog "
    Else
        terms = Script.toVBAArray( prop.blobTermsValue)
        For i = 0 To UBound( terms)
            terms( i) = 0
        Next
        prop.blobTermsValue = Script.toCPPArray( terms)
        If iv.applySettings( prop) <> 0 Then
            Client.writeToLog "
        Else
            MsgBox "
        End If
    End If
End If
```

1.2.3.4.7 Метод toCPPArray

Метод предназначен для преобразования массивов, используемых в VBScript, в формат массивов, принятых в системе APACS Bio.

**VBScript**

toCPPArray( vbArray)  
где vbArray — массив в формате, принятом в VBScript.

**Возвращаемое значение**

В случае успешного выполнения метод возвращает массив в формате, принятом в системе APACS Bio.

**Пример**

```
'
iv

If Apacs.GetObjectByAlias( "iv", iv) <> 0 Then
    Client.writeToLog "iv"
Else
    If iv.GetCurrentSettings( prop) <> 0 Then
        Client.writeToLog "
    Else
        terms = Script.toVBAArray( prop.blobTermsValue)
        For i = 0 To UBound( terms)
            terms( i) = 0
        Next
        prop.blobTermsValue = Script.toCPPArray( terms)
        If iv.applySettings( prop) <> 0 Then
            Client.writeToLog "
        Else
            MsgBox "
        End If
    End If
End If
```

```

Next
prop.blobTermsValue = Script.toCPPArray( terms)
If iv.applySettings( prop) <> 0 Then
    Client.writeToLog "
Else
    MsgBox "
End If
End If
End If

```

#### 1.2.3.4.8 Метод loadCPPArray

Метод предназначен для загрузки в переменную APACS Bio массива из указанного файла.

##### **VBScript**

loadCPPArray( strFile, cppArray)

где strFile — путь к внешнему файлу,

cppArray — массив в формате, принятом в системе APACS Bio.

##### **Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

##### **Возможные причины невыполнения**

- ошибка работы с файловой системой.

##### **Пример сохранения фотографии владельца на диск и загрузка другой фотографии**

```

'                                     hld                                     jpeg

If Apacs.getObjectByAlias( "hld", hld) <> 0 Then
    Client.writeToLog "                                     hld"
Else
    If hld.getChildrenObjsByTypes( Array( "TApcCHMainPhoto") , photos) <> 0 Then
        Client.writeToLog "
    Else
        If UBound( photos) < 0 Then
            Client.writeToLog "
        Else
            If photos( 0) .getCurrentSettings( prop) <> 0 Then
                Client.writeToLog "
            Else
                If Script.saveCPPArray( "C:\Documents and Settings\All Users\Application Data\photosrc.jpg",
                    prop.binBufPhoto) <> 0 Then
                    Client.writeToLog "
                Else
                    If Script.loadCPPArray( "C:\Documents and Settings\All Users\Application Data\photodst.jpg",
                        blobPhoto) <> 0 Then
                        Client.writeToLog "
                    Else
                        prop.binBufPhoto = blobPhoto
                        If photos( 0) .applySettings( prop) <> 0 Then
                            Client.writeToLog "
                        Else
                            MsgBox "
                        End If
                    End If
                End If
            End If
        End If
    End If
End If

```

```
End If
End If
End If
```

#### 1.2.3.4.9 Метод saveCPPArray

Метод предназначен для сохранения в указанный файл массива в формате APACS Bio.

##### **VBScript**

saveCPPArray( strFile, cppArray) ,  
где strFile — путь к внешнему файлу,  
cppArray — массив в формате, принятом в системе APACS Bio.

##### **Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

##### **Возможные причины невыполнения**

- ошибка работы с файловой системой.

##### **Пример сохранения фотографии владельца на диск и загрузка другой фотографии**

```
'                                     hld                                     jpeg

If Apacs.GetObjectByAlias( "hld", hld) <> 0 Then
    Client.writeToLog "                                     hld"
Else
    If hld.ChildrenObjsByTypes( Array( "TApcCHMainPhoto") , photos) <> 0 Then
        Client.writeToLog "                                     "
    Else
        If UBound( photos) < 0 Then
            Client.writeToLog "                                     "
        Else
            If photos( 0) .GetCurrentSettings( prop) <> 0 Then
                Client.writeToLog "                                     "
            Else
                If Script.saveCPPArray( "C:\Documents and Settings\All Users\Application Data\photosrc.jpg",
                    prop.binBufPhoto) <> 0 Then
                    Client.writeToLog "                                     "
                Else
                    If Script.loadCPPArray( "C:\Documents and Settings\All Users\Application Data\photodst.jpg",
                        blobPhoto) <> 0 Then
                        Client.writeToLog "                                     "
                    Else
                        prop.binBufPhoto = blobPhoto
                        If photos( 0) .applySettings( prop) <> 0 Then
                            Client.writeToLog "                                     "
                        Else
                            MsgBox "                                     "
                        End If
                    End If
                End If
            End If
        End If
    End If
End If
End If
End If
End If
```

### 1.2.3.5 Интерфейс IApcObjectWrap

Интерфейс объекта конфигурации в системе APACS Bio. Предоставляет следующие методы:

Метод	Описание
<a href="#"><u>getCurrentSettings</u></a>	Метод позволяет получить текущие настройки объекта.
<a href="#"><u>applySettings</u></a>	Метод позволяет сохранить на сервере APACS Bio переданный список измененных параметров объекта.
<a href="#"><u>getChildrenObjsByTypes</u></a>	Метод предназначен для получения списка дочерних объектов указанного типа, существующих на текущий момент в конфигурации системы у данного объекта.
<a href="#"><u>getApacsType</u></a>	Метод позволяет получить строковый идентификатор типа данного объекта.
<a href="#"><u>getChildrenObjs</u></a>	Метод позволяет получить список всех дочерних объектов данного объекта, существующих на данный момент в конфигурации системы.
<a href="#"><u>getParentObject</u></a>	Метод предназначен для получения родительского объекта указанного объекта.
<a href="#"><u>getUID</u></a>	Метод позволяет получить строковое представление идентификатора данного объекта.
<a href="#"><u>getChildSettingsForAdd</u></a>	Метод позволяет получить настройки по умолчанию дочернего объекта для его добавления в систему.
<a href="#"><u>addChildWithSettings</u></a>	Метод позволяет к данному объекту добавить дочерний объект с указанными настройками.
<a href="#"><u>deleteObject</u></a>	Метод предназначен для удаления объекта из конфигурации системы.
<a href="#"><u>getEventSettingsForRegister</u></a>	Метод позволяет получить настройки по умолчанию для регистрации нового сообщения в системе.
<a href="#"><u>registerEventWithSettings</u></a>	Метод позволяет зарегистрировать в системе новое сообщение от данного объекта.

<u>&lt;команда&gt;</u>	Метод предназначен для выполнения команд управления, поддерживаемых объектом.
------------------------	---

1.2.3.5.1 Метод <команда>

Метод предназначен для выполнения команд управления, поддерживаемых объектом.

VBScript

< > ( 1, ... N)  
где < > — идентификатор команды,  
1, ... N — параметры выполнения команды.

**Возвращаемое значение**  
В случае успешного выполнения метод возвращает 0.

- Возможные причины невыполнения**
- нет связи с сервером;
  - объект удален;
  - нет возможности выполнить команду;
  - нет прав на выполнение команд этого объекта.

**Пример выполнения команды *Заккрыть* на объекте с псевдонимом reader**

```
If Apacs.GetObjectByAlias( "reader", Obj) <> 0 Then
    Client.writeToLog "reader"
Else
    If Obj.cmdLock( ) <> 0 Then
        Client.writeToLog "
    Else
        msgbox "
    End If
End If
```

1.2.3.5.2 Метод getCurrentSettings

Метод позволяет получить текущие настройки объекта.

VBScript

getCurrentSettings( PropObj)  
где PropObj — возвращаемое значение, текущие параметры объекта.

**Возвращаемое значение**  
В случае успешного выполнения метод возвращает 0.

- Возможные причины невыполнения:**
- нет связи с сервером;
  - объект удален.

**Пример получения имени корня системы**

```
If Apacs.getRootObject( objRoot) <> 0 Then
    Client.writeToLog "
Else
    If objRoot.getCurrentSettings( prop) <> 0 Then
        Client.writeToLog "
    Else
        msgbox " : " + prop.strName
```



```
End If  
End If
```

### C++

```
HRESULT getCurrentSettings( LPDISPATCH  
                           appSettings/*[out]*/,  
                           int* apnResult/*[out,retval]*/)
```

#### Параметры

[out] appSettings — текущие настройки объекта,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

#### Возвращаемое значение

Возвращает значение HRESULT.

### 1.2.3.5.3 Метод applySettings

Метод позволяет сохранить на сервере APACS Bio переданный список измененных параметров объекта.

### VBScript

```
applySettings( PropObj)  
где PropObj — список новых параметров объекта.
```

#### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

#### Возможные причины невыполнения

- нет связи с сервером,
- объект удален,
- введены некорректные значения параметров,
- нет прав на редактирование объекта.

#### Пример изменения свойства объекта

```
,  
If Apacs.getRootObject( objRoot) <> 0 Then  
    Client.writeToLog " "  
Else  
    If objRoot.getCurrentSettings( prop) <> 0 Then  
        Client.writeToLog " "  
    Else  
        prop.strName = " "  
        If objRoot.applySettings( prop) <> 0 Then  
            Client.writetoLog " "  
        Else  
            msgbox " "  
        End If  
    End If  
End If  
End If
```

### C++

```
HRESULT applySettings( LPDISPATCH apSettings/*[in]*/,  
                      int* apnResult/*[out,retval]*/)
```

#### Параметры

[in] apSettings — новые настройки объекта,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

**1.2.3.5.4 Метод getChildrenObjs**

Метод позволяет получить список всех дочерних объектов данного объекта, существующих на данный момент в конфигурации системы.

**VBScript**

getChildrenObjs( arrObjs)

где arrObjs — возвращаемое значение, список всех дочерних объектов данного объекта, существующих на данный момент в конфигурации системы.

**Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

**Возможные причины невыполнения**

- нет связи с сервером,
- объект удален.

**Пример получения имен дочерних элементов корня**

```
If Apacs.GetRootObject( objRoot) <> 0 Then
    Client.writeToLog "
Else
    If objRoot.getChildrenObjs( chlds) <> 0 Then
        Client.writeToLog "
    Else
        first = true
        strChlds = ""
        For i = 0 To UBound ( chlds)
            If chlds( i) .GetCurrentSettings( prop) = 0 Then
                If first Then
                    strChlds = prop.strName
                    first = false
                Else
                    strChlds = strChlds + vbNewLine + prop.strName
                End If
            End If
        Next
        MsgBox " : " + vbNewLine + strChlds
    End If
End If
```

**C++**

```
HRESULT getChildrenObjs( LPSAFEARRAY* apChildrenObjs/*[out]*/,
                        int* apnResult/*[out,retval]*/)
```

**Параметры**

[out] apChildrenObjs — массив объектов, поддерживающих интерфейс [IApcObjectWrap](#),

[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

### 1.2.3.5.5 Метод getChildrenObjsByTypes

Метод предназначен для получения списка дочерних объектов указанного типа, существующих на текущий момент в конфигурации системы у данного объекта.

#### VBScript

getChildrenObjsByTypes( arrTypes, arrObjs)

где arrTypes — список типов дочерних объектов,  
arrObjs — возвращаемое значение, список дочерних объектов указанного типа, существующих на текущий момент в конфигурации системы у данного объекта.

#### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

#### Возможные причины невыполнения

- нет связи с сервером,
- объект удален.

#### Пример получения имен всех папок, прикрепленных к корню

```
If Apacs.getRootObject( objRoot) = 0 Then
  Client.writeToLog "
  If objRoot.getChildrenObjsByTypes( Array( "TApcFolder") , chlds) <> 0 Then
    Client.writeToLog "
  Else
    first = true
    strChlds = ""
    For i = 0 To UBound ( chlds)
      If chlds( i) .getCurrentSettings( prop) = 0 Then
        If first Then
          strChlds = prop.strName
          first = false
        Else
          strChlds = strChlds + vbNewLine + prop.strName
        End If
      End If
    Next
    MsgBox " : " + vbNewLine + strChlds
  End If
End If
```

#### C++

```
HRESULT getChildrenObjsByTypes( LPSAFEARRAY aObjTypes/*[in]*/,
                                LPSAFEARRAY* apChildrenObjs/*[out]*/,
                                int* apnResult/*[out,retval]*/)
```

#### Параметры

[in] aObjTypes — список типов дочерних объектов,  
[out] apChildrenObjs — массив объектов, поддерживающих интерфейс [IApcObjectWrap](#),  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

#### Возвращаемое значение

Возвращает значение HRESULT.

### 1.2.3.5.6 Метод getParentObject

Метод предназначен для получения родительского объекта указанного объекта.

#### VBScript

```
getParentObject( parentObj)
```

где parentObj — возвращаемое значение, родительский объект указанного объекта.

#### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

#### Возможные причины невыполнения

- нет связи с сервером,
- объект удален,
- объект является *Корневым элементом системы*.

#### Пример получения имени родительского объекта для объекта с псевдонимом contr

```
If Apacs.GetObjectByAlias( "contr", obj) <> 0 Then
    Client.writeToLog "                                contr"
Else
    If obj.getParentObject( objParent) <> 0 Then
        Client.writeToLog "                                "
    Else
        If objParent.getCurrentSettings( prop) = 0 Then
            msgbox "                                : " + prop.strName
        End If
    End If
End If
```

#### C++

```
HRESULT getParentObject( LPDISPATCH* appObject/*[out]*/,
                        int* apnResult/*[out,retval]*/)
```

#### Параметры

[out] appObject — указатель на родительский объект, поддерживающий интерфейс [IApcObjectWrap](#),

[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

#### Возвращаемое значение

Возвращает значение HRESULT.

### 1.2.3.5.7 Метод deleteObject

Метод позволяет удалить объект из конфигурации системы.

#### VBScript

```
deleteObject( )
```

#### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

#### Возможные причины невыполнения

- нет связи с сервером,
- нет прав на удаление объекта,
- объект не существует в системе.

**Пример удаления объекта с псевдонимом obj4del**

```
If Apacs.GetObjectByAlias( "obj4del", obj) <> 0 Then
    Client.writeToLog "                obj4del"
Else
    If obj.deleteObject( ) <> 0 Then
        Client.writeToLog "                "
    Else
        msgbox "                "
    End If
End If
```

**C++**

```
HRESULT deleteObject( int* apnResult/*[out,retval]*/)
```

**Параметры**

[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

**1.2.3.5.8 Метод getChildSettingsForAdd**

Метод позволяет получить настройки по умолчанию дочернего объекта для его добавления в систему.

**VBScript**

```
getChildSettingsForAdd ( strObjType, propObj)
```

где strObjType — тип добавляемого объекта,

propObj — возвращаемое значение, настройки нового объекта по умолчанию.

**Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

**Возможные причины невыполнения**

- нет связи с сервером,
- объектов такого типа больше нельзя добавить в конфигурацию системы (существуют ограничения по конфигурированию либо ограничения в лицензии).

**Пример добавления объекта типа Папка**

```
If Apacs.getRootObject( ObjRoot) <> 0 Then
    Client.writeToLog "                "
Else
    If objRoot.getChildSettingsForAdd( "TApcFolder", prop) <> 0 Then
        Client.writeToLog "                "
    Else
        prop.strName = "                "
        If objRoot.addChildWithSettings( prop, objFld) <> 0 Then
            Client.writeToLog "                "
        Else
            msgbox "                "
        End If
    End If
End If
```

**C++**

```
HRESULT getChildSettingsForAdd( BSTR aObjType/*[in]*/,
                                LPDISPATCH* apSettings/*[out]*/,
                                int* apnResult/*[out,retval]*/)
```

### Параметры

[in] aObjType — тип добавляемого объекта,  
 [out] apSettings — настройки нового объекта по умолчанию,  
 [out] apnResult — успешное выполнение — 0, иначе — код ошибки.

### Возвращаемое значение

Возвращает значение HRESULT.

#### 1.2.3.5.9 Метод addChildWithSettings

Метод позволяет к данному объекту добавить дочерний объект с указанными настройками.

### VBScript

```
addChildWithSettings( propObj, newObj)
```

где propObj — настройки добавляемого объекта,  
 newObj — возвращаемое значение, добавленный объект.

### Возвращаемое значение

В случае успешного выполнения метод возвращает 0.

### Возможные причины невыполнения

- нет связи с сервером,
- нет прав на добавление объекта,
- некорректные значения свойств объекта.

### Пример добавления объекта типа *Папка*

```
If Apacs.getRootObject( ObjRoot) <> 0 Then
    Client.writeToLog "
Else
    If objRoot.getChildSettingsForAdd( "TApcFolder", prop) <> 0 Then
        Client.writeToLog "
    Else
        prop.strName = "
        If objRoot.addChildWithSettings( prop, objFld) <> 0 Then
            Client.writeToLog "
        Else
            msgbox "
        End If
    End If
End If
End If
```

### C++

```
HRESULT addChildWithSettings( LPDISPATCH apSettings/*[in]*/,
                                LPDISPATCH* appObject/*[out]*/,
                                int* apnResult/*[out,retval]*/)
```

### Параметры

[in] apSettings — настройки добавляемого объекта,  
 [out] appObject — указатель на новый объект, поддерживающий интерфейс [IApcObjectWrap](#),  
 [out] apnResult — успешное выполнение — 0, иначе — код ошибки.

### Возвращаемое значение

Возвращает значение HRESULT.

1.2.3.5.10 Метод `getEventSettingsForRegister`

Метод позволяет получить настройки по умолчанию для регистрации нового сообщения в системе.

**VBScript**

`getEventSettingsForRegister( strEventType, propEvent)`  
где `strEventType` — тип регистрируемого сообщения,  
      `propEvent` — возвращаемое значение, настройки нового сообщения по умолчанию.

**Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

**Возможные причины невыполнения**

- нет связи с сервером.

**Пример регистрации сообщения *Доступ разрешен***

```
'
' 1.                                rdr
' 2.                                1000                                crd
' 3.                                c                                hld
'                                1000

If Apacs.GetObjectByAlias( "rdr", rdr) <> 0 Then
    Client.writeToLog "                                rdr"
Else
    If Apacs.GetObjectByAlias( "crd", crd) <> 0 Then
        Client.writeToLog "                                crd"
    Else
        If Apacs.GetObjectByAlias( "hld", hld) <> 0 Then
            Client.writeToLog "                                hld"
        Else
            If rdr.getEventSettingsForRegister( "TApcCardHolderAccess__Granted", prop) <> 0 Then
                Client.writeToLog "                                "
            Else
                prop.isOffLineEvent = false
                prop.dwCardNumber = 1000
                set prop.SysAddrCard = crd
                prop.strHolderName = "                                ."
                set prop.SysAddrHolder = hld
                If rdr.registerEventWithSettings( prop) <> 0 Then
                    Client.writeToLog "                                "
                Else
                    msgbox "                                "
                End If
            End If
        End If
    End If
End If
```

**C++**

`HRESULT getEventSettingsForRegister( BSTR aEventType/*[in]*/,  
 LPDISPATCH* appSettings/*[out]*/,  
 int* apnResult/*[out,retval]*/)`

**Параметры**

`[in] aEventType` — тип регистрируемого сообщения,

[out] appSettings — настройки нового сообщения по умолчанию,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**  
Возвращает значение HRESULT.

1.2.3.5.11 Метод registerEventWithSettings

Метод позволяет зарегистрировать в системе новое сообщение от данного объекта.

**VBScript**

registerEventWithSettings( propEvent)  
где propEvent — настройки регистрируемого сообщения.

**Возвращаемое значение**  
В случае успешного выполнения метод возвращает 0.

**Возможные причины невыполнения**  
• нет связи с сервером.

**Пример регистрации сообщения *Доступ разрешен***

```
'
' 1.                rdr
' 2.                1000                crd
' 3.                c                hld
'                1000

If Apacs.GetObjectByAlias( "rdr", rdr) <> 0 Then
    Client.writeToLog "                rdr"
Else
    If Apacs.GetObjectByAlias( "crd", crd) <> 0 Then
        Client.writeToLog "                crd"
    Else
        If Apacs.GetObjectByAlias( "hld", hld) <> 0 Then
            Client.writeToLog "                hld"
        Else
            If rdr.getEventSettingsForRegister( "TApcCardHolderAccess__Granted", prop) <> 0 Then
                Client.writeToLog "                "
            Else
                prop.isOffLineEvent = false
                prop.dwCardNumber = 1000
                set prop.SysAddrCard = crd
                prop.strHolderName = "                ."
                set prop.SysAddrHolder = hld
                If rdr.registerEventWithSettings( prop) <> 0 Then
                    Client.writeToLog "                "
                Else
                    msgbox "                "
                End If
            End If
        End If
    End If
End If
```

**C++**

HRESULT registerEventWithSettings( LPDISPATCH appSettings/\*[in]\*/,  
int\* apnResult/\*[out,retval]\*/)

**Параметры**



[in] appSettings — настройки регистрируемого сообщения,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

**1.2.3.5.12 Метод getUID**

Метод позволяет получить строковое представление идентификатора данного объекта.

**VBScript**

getUID( strUID)

где strUID — возвращаемое значение, идентификатор объекта.

Формат возвращаемого значения (strUID) имеет следующий вид: SA XXXX.YYYYYYYY, где XXXX – первая часть системного адреса в шестнадцатеричном формате, а YYYYYYYY – вторая часть системного адреса в шестнадцатеричном формате.

**Возвращаемое значение**

В случае успешного выполнения метод возвращает 0.

**Возможные причины невыполнения**

- нет связи с сервером,
- объект удален.

**Пример получения идентификатора объекта**

```
If Apacs.getRootObject( objRoot) <> 0 Then
    Client.writeToLog "
Else
    If objRoot.getUID( strUID) <> 0 Then
        Client.writeToLog "
    Else
        msgbox " = " + strUID
    End If
End If
```

**C++**

```
HRESULT getUID( BSTR* astrUID/*[out]*/,
                int* apnResult/*[out,retval]*/)
```

**Параметры**

[out] astrUID — идентификатор объекта,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

**1.2.3.5.13 Метод getApacsType**

Метод позволяет получить строковый идентификатор типа данного объекта.

**VBScript**

getApacsType( strType)

где strType — возвращаемое значение, тип объекта в системе.

**Возвращаемое значение**  
В случае успешного выполнения метод возвращает 0.

- Возможные причины невыполнения**
- нет связи с сервером,
  - объект удален.

**Пример определения типа объекта**

```
If Apacs.getRootObject( objRoot) <> 0 Then
    Client.writeToLog "
Else
    If objRoot.getApacsType( strType) <> 0 Then
        Client.writeToLog "
    Else
        msgbox "          = " + strType
    End If
End If
```

C++

```
HRESULT getApacsType( BSTR* astrApacsType/*[out]*/,
                    int* apnResult/*[out,retval]*/)
```

**Параметры**  
[out] astrApacsType — тип объекта,  
[out] apnResult — успешное выполнение — 0, иначе — код ошибки.

**Возвращаемое значение**  
Возвращает значение HRESULT.

1.2.3.6 Интерфейс IApcVideoLinkerWrap

Интерфейс объекта *VideoLinker*. Предоставляет метод:

Метод	Описание
<a href="#">linkRec</a>	Метод позволяет инициировать запись на видеокамере и связать видеофрагмент с поступившим сообщением.

1.2.3.6.1 Метод linkRec

Метод позволяет инициировать запись на видеокамере и связать видеофрагмент с поступившим сообщением. В момент записи видеофрагмента в системе регистрируется дочернее сообщение типа *Видеофрагмент связан с сообщением*.

VBScript

```
linkRec ( Event, CamObj, dwTime)
```

где Event — сообщение, для которого была инициирована запись видео,  
CamObj — камера, производящая запись видеофрагмента,  
dwTime — продолжительность записи видеофрагмента в секундах.

**Возвращаемое значение**  
В случае успешного выполнения метод возвращает 0.

- Возможные причины невыполнения**
- нет связи с сервером,

- нет прав на просмотр объекта *Камера*.

**Пример записи видеосфрагмента**

```
Sub Apacs_Event ( evt)
  If Apacs.GetObjectByAlias( "Camera", obj ) = 0 Then
    VideoLinker.LinkRec evt, obj, 5
  End If
End Sub
```

**1.2.3.7 Интерфейс IApcEQUALObjFilter**

Интерфейс *IApcEQUALObjFilter*, предназначен для создания фильтра, проверяющего эквивалентность. Имеет следующие свойства.

Свойство	Тип	Описание
strName	string	Строка, определяющая свойства объекта или события в APACS Bio.
Value	variant	Значение свойства, имя которого определено в strName.

Пример использования интерфейса *IApcEQUALObjFilter* смотрите в описании метода [getObjectsByFilter](#).

**1.2.3.8 Интерфейс IApcANDObjFilter**

Интерфейс *IApcANDObjFilter*, предназначен для объединения фильтров по логическому И.

Метод	Описание
<a href="#">addCondition</a>	Метод позволяет добавить фильтр.
<a href="#">getConditions</a>	Метод позволяет получить список фильтров, которые объединяются по логическому И.

Пример использования интерфейса *IApcANDObjFilter* смотрите в описании метода [getObjectsByFilter](#).

**1.2.3.8.1 Метод addCondition**

Метод позволяет добавить фильтр.

**VBScript**

```
addCondition ( Condition)
где Condition — экземпляр объекта, реализующего интерфейс IApcANDObjFilter или IApcEQUALObjFilter.
```

**C++**

```
HRESULT addCondition( LPUNKNOWN apCondition/*[in]*/)

```

**Параметры**

[in] apCondition — экземпляр объекта, реализующего интерфейс IApcANDObjFilter или IApcEQUALObjFilter.

**Возвращаемое значение**  
Возвращает значение HRESULT.

1.2.3.8.2 Метод getConditions

Метод позволяет получить список фильтров, которые объединяются по логическому И.

VBScript

getConditions

**Возвращаемое значение**  
В случае успешного выполнения метод возвращает список фильтров, объединенных по логическому условию.

C++

HRESULT **getConditions**( LPSAFEARRAY\* apConditions/\*[out,retval]\*/)

**Параметры**  
[out,retval] apConditions — возвращаемое значение, список фильтров, объединенных по логическому условию.

**Возвращаемое значение**  
Возвращает значение HRESULT.

1.2.3.9 Интерфейс IApcHelper

Интерфейс предназначен для работы с данными, запакованными в BLOB поля: с картами, владельцами карт и группами доступа.

объекта в системе APACS Bio. Предоставляет следующие методы:

Метод	Описание
<a href="#">setServer</a>	Метод позволяет получить доступ к серверу для дальнейшей работы с данными.
<a href="#">getAccount</a>	Метод позволяет получить данные карты.
<a href="#">applyAccount</a>	Метод позволяет записать в BLOB данные карты.
<a href="#">getCardHolder</a>	Метод позволяет получить данные владельца карты.
<a href="#">applyCardHolder</a>	Метод позволяет записать в BLOB данные владельца карты.
<a href="#">getSecurityGroup</a>	Метод позволяет получить данные группы доступа.
<a href="#">applySecurityGroup</a>	Метод позволяет записать в BLOB данные группы доступа.

1.2.3.9.1 Метод setServer

Метод позволяет получить доступ к серверу для дальнейшей работы с данными.

VBScript

setServer(Apcsrvsdk\_tlb::IApcServerWrap\* server/\*[in]\*/)

#### 1.2.3.9.2 Метод getAccount

Метод позволяет получить из блока данные карты.

##### **VBScript**

```
getAccount(LPDISPATCH props/*[in]*/,  
           Apcsdsecuritymanagerhelper_tlb::IApcAccountWrap* account/*[out,retval]*/)
```

#### 1.2.3.9.3 Метод applyAccount

Метод позволяет записать в BLOB данные карты.

##### **VBScript**

```
applyAccount(LPDISPATCH props/*[in]*/,  
             Apcsdsecuritymanagerhelper_tlb::IApcAccountWrap* account/*[in]*/)
```

#### 1.2.3.9.4 Метод getCardHolder

Метод позволяет получить из блока данные владельца карты.

#### 1.2.3.9.5 Метод applyCardHolder

Метод позволяет записать в BLOB данные владельца карты.

#### 1.2.3.9.6 Метод getSecurityGroup

Метод позволяет получить из блока данные группы доступа.

#### 1.2.3.9.7 Метод applySecurityGroup

Метод позволяет записать в BLOB данные группы доступа.

### 1.3 Клиентский модуль HTML обозреватель

Клиентский модуль **HTML обозреватель** позволяет создавать пользовательский интерфейс на основе загруженных HTML документов, объектная модель которых расширена объектами APACS Bio.

Объектная модель APACS Bio доступна из объекта DHTML window.external:

- *Apacs* — window.external.Apacs,
- *Log* — window.external.Log,
- *Client* — window.external.Client,
- *Script* — window.external.Script,
- *VideoLinker* - window.external.VideoLinker.

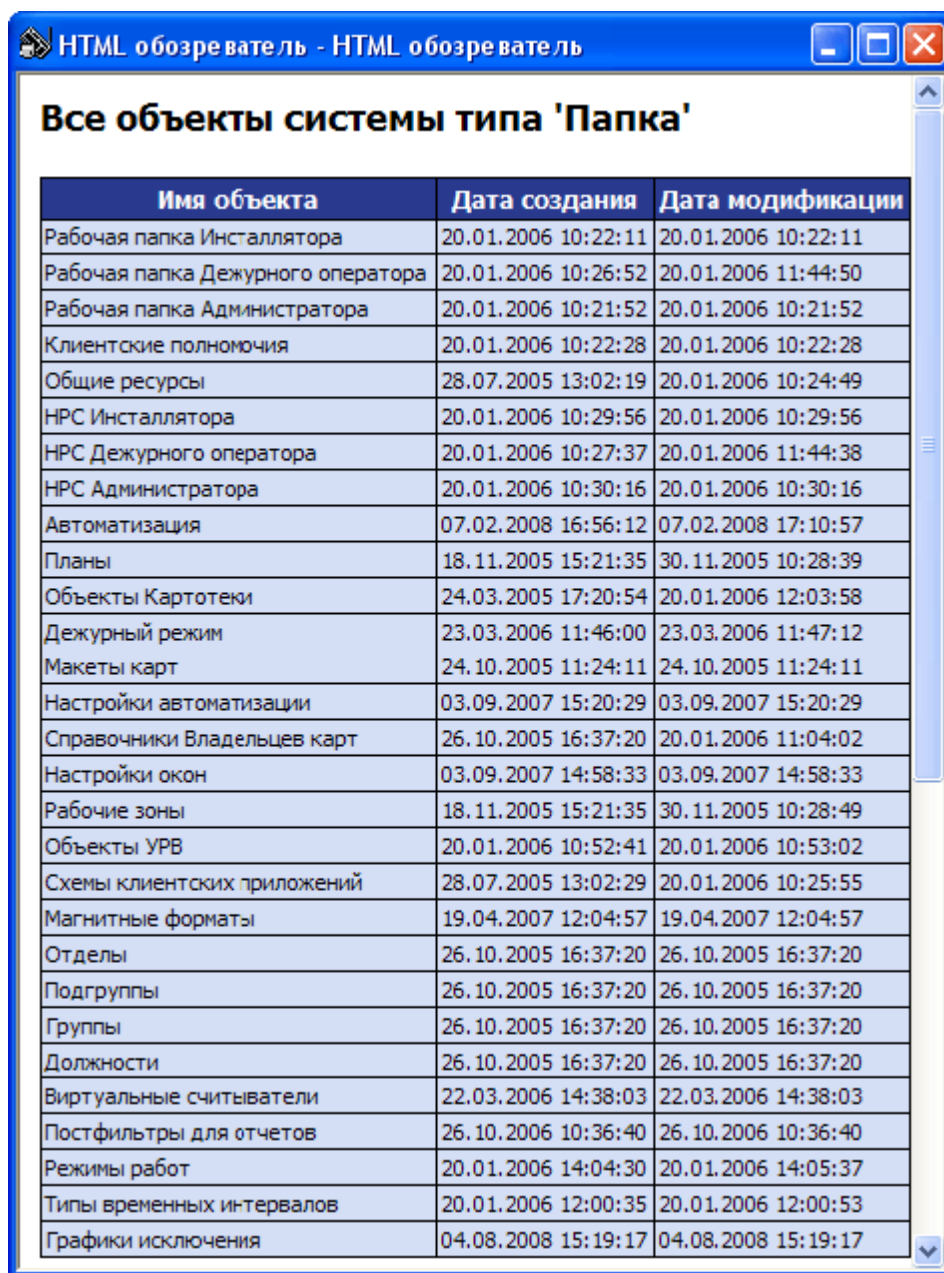
Таким образом, администратор комплекса может написать собственные HTML документы, используя весь доступный функционал скриптов. С помощью HTML документов могут быть реализованы такие функции, как:

- отчет об объектах и сообщениях системы,
- пропуск людей, предъявивший карту на считывателе, по решению оператора,
- интерактивное управление оборудованием и др.

В результате комплекс легко настраивается под требования конкретной системы управления и контроля доступом.

**Обратите внимание:** для корректной работы модуля **HTML обозреватель** в настройках Вашего Internet Explorer Browser требуется разрешить выполнение сценариев (скриптов).

Для загрузки в систему HTML документов используются объекты типа **HTML обозреватель**.



Имя объекта	Дата создания	Дата модификации
Рабочая папка Инсталлятора	20.01.2006 10:22:11	20.01.2006 10:22:11
Рабочая папка Дежурного оператора	20.01.2006 10:26:52	20.01.2006 11:44:50
Рабочая папка Администратора	20.01.2006 10:21:52	20.01.2006 10:21:52
Клиентские полномочия	20.01.2006 10:22:28	20.01.2006 10:22:28
Общие ресурсы	28.07.2005 13:02:19	20.01.2006 10:24:49
НПС Инсталлятора	20.01.2006 10:29:56	20.01.2006 10:29:56
НПС Дежурного оператора	20.01.2006 10:27:37	20.01.2006 11:44:38
НПС Администратора	20.01.2006 10:30:16	20.01.2006 10:30:16
Автоматизация	07.02.2008 16:56:12	07.02.2008 17:10:57
Планы	18.11.2005 15:21:35	30.11.2005 10:28:39
Объекты Картотеки	24.03.2005 17:20:54	20.01.2006 12:03:58
Дежурный режим	23.03.2006 11:46:00	23.03.2006 11:47:12
Макеты карт	24.10.2005 11:24:11	24.10.2005 11:24:11
Настройки автоматизации	03.09.2007 15:20:29	03.09.2007 15:20:29
Справочники Владельцев карт	26.10.2005 16:37:20	20.01.2006 11:04:02
Настройки окон	03.09.2007 14:58:33	03.09.2007 14:58:33
Рабочие зоны	18.11.2005 15:21:35	30.11.2005 10:28:49
Объекты УРВ	20.01.2006 10:52:41	20.01.2006 10:53:02
Схемы клиентских приложений	28.07.2005 13:02:29	20.01.2006 10:25:55
Магнитные форматы	19.04.2007 12:04:57	19.04.2007 12:04:57
Отделы	26.10.2005 16:37:20	26.10.2005 16:37:20
Подгруппы	26.10.2005 16:37:20	26.10.2005 16:37:20
Группы	26.10.2005 16:37:20	26.10.2005 16:37:20
Должности	26.10.2005 16:37:20	26.10.2005 16:37:20
Виртуальные считыватели	22.03.2006 14:38:03	22.03.2006 14:38:03
Постфильтры для отчетов	26.10.2006 10:36:40	26.10.2006 10:36:40
Режимы работ	20.01.2006 14:04:30	20.01.2006 14:05:37
Типы временных интервалов	20.01.2006 12:00:35	20.01.2006 12:00:53
Графики исключения	04.08.2008 15:19:17	04.08.2008 15:19:17

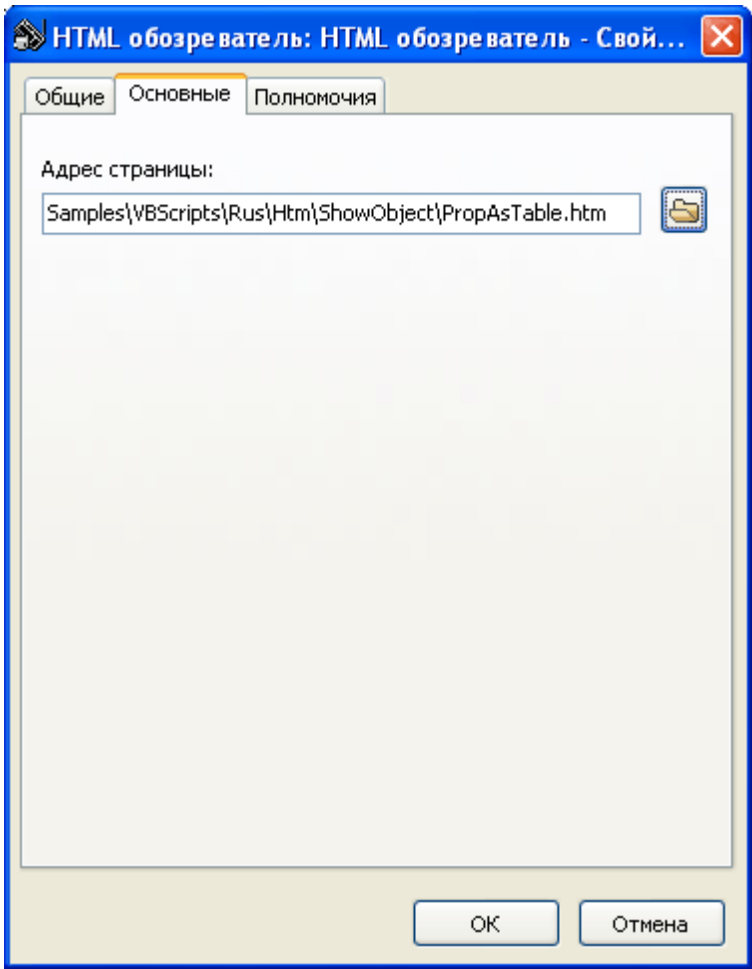
Окно HTML документа с отчетом обо всех объектах типа *Папка*

### **Конфигурирование объекта HTML обозреватель**

Конфигурирование объекта HTML обозреватель осуществляется в окне **Проводник**:

- В окне **Проводник** добавьте объект типа **HTML обозреватель** к объекту типа **Папка**. Откроется окно **Файл – Свойства**. Укажите название объекта и нажмите кнопку **ОК**. Объект появится в дереве объектов окна **Проводник**.
- Сконфигурируйте объект **HTML обозреватель**. Для этого выполните на объекте команду **Редактировать**. На вкладке «**Основные**» этого

объекта в поле **Адрес страницы** требуется указать путь к локальному HTML документу или URL HTML страницу.



Вкладка «Основные» окна редактирования свойств объекта **HTML обозреватель**

**Просмотр HTML документа**

Чтобы открыть на экране окно HTML документа, выделите объект в окне **Проводник** и воспользуйтесь командой **Показать**.

**1.3.1 Редактор HTML обозревателя**

**Файл** типа **HTML обозреватель** предназначен для загрузки в систему написанных администратором HTML документов, объектная модель которых расширена объектами APACS Bio.

Настройки

Все настройки объекта находятся на следующих вкладках:

- Общие
- Основные
- Полномочия

На вкладке «**Основные**» находятся следующие настройки:

- **Адрес страницы** — укажите в этом поле путь к локальному HTML документу или URL HTML страницы.

Клиентские команды	Описание
--------------------	----------

<b>Показать</b>	С помощью этой команды можно открыть на экране окно с HTML документом.
<b>Свойства файла</b>	С помощью этой команды можно открыть окно свойств файла.

1.4 COM SDK APACS 3000 (комплект для разработки программного обеспечения)

COM SDK (комплект для разработки программного обеспечения) представляет собой библиотеку, с помощью которой сторонние разработчики могут вызывать функции серверных объектов ПК APACS Bio и, таким образом, осуществлять интеграцию с комплексом или же разрабатывать собственные клиентские приложения. Если при помощи подсистемы автоматизации создаются скрипты, которые исполняются внутри приложений APACS Bio, то SDK предоставляет возможность вызывать функции и прослушивать сообщения APACS Bio из внешних программ.

COM SDK предоставляет следующие возможности:
авторизация на сервере APACS Bio
доступ к любым объектам конфигурации
получение текущих настроек объектов
редактирование настроек объектов
создание новых объектов
удаление объектов
получение сообщений от объектов в режиме online
получение уведомлений об удалении, изменении, добавлении объектов в режиме online
получение сохраненных сообщений из базы данных по фильтру
регистрация сообщений от объектов
получение уведомления в случае потери связи с сервером

Функции COM SDK реализованы через COM-объект, находящийся в библиотеке ArcSrvConnection.dll. Регистрация COM-объекта происходит автоматически при инсталляции комплекса. Библиотека предоставляет все необходимые COM-объекты для прямого вызова функций сервера. Так как COM SDK построен на основе технологии COM, возможно использовать SDK APACS Bio из любых сред программирования, которые поддерживают работу с COM (например, MS Visual C++, MS Visual C#, C++ Builder, Delphi и др.). Также работу с COM SDK возможно реализовать через стандартный \*.vbs файл, который может быть разработан в любом редакторе и запущен из командной строки операционной системы.

В составе комплекса поставляются примеры на языке C++ Builder, VBScript и Visual C# для платформы .Net Framework 1.1 (находятся в папке Samples \SDK\_COM корневого каталога APACS Bio). Примеры демонстрируют все основные возможности COM SDK.

COM SDK APACS Bio имеет следующие ограничения:
COM SDK предоставляет возможность вызова серверных команд



комплекса, но не позволяет встраивать вызов какого-либо дополнительного функционала «внутри» уже существующих клиентских приложений (например, нельзя дописать функционал модулей <b>Редактор планов, Просмотр планов, Проходная</b> и т.д.).
Для получения сообщений из базы данных доступны только запросы с фильтром по дате начала, дате окончания и типу событий.
Получение списка объектов типа <i>Владелец карты, Группа доступа, Идентификатор</i> возможно только в несортированном списке, без возможности задания фильтрации (или же по псевдонимам, указанным для каждого объекта).
Через COM SDK недоступен объект «Метаданные» (информация о типах, полях и т.д.).
Через COM SDK нельзя получить переводы для информации в «Метаданных» (например, стандартные названия типов считывателей, входов и т.д.).
При помощи COM SDK нельзя читать/редактировать настройки полномочий для объектов конфигурации.
При помощи COM SDK нельзя разработать серверный драйвер оборудования, работающий в составе сервера APACS Bio (так называемый DDK).

**Принципы работы**

Для того чтобы начать работу с сервером APACS Bio, необходимо установить с ним соединение. Для этого предназначен класс *TApcConnection*. Класс *TApcConnection* реализует метод [createSession](#), позволяющий Вам создать новую сессию соединения с сервером APACS Bio.

Класс *TApcSession* предоставляет сеанс связи с сервером APACS Bio. Из него доступен объект с интерфейсом [IApcServerWrap](#), который отображает объектную модель сервера APACS Bio.

Для получения подробной информации о COM SDK APACS Bio смотри описание интерфейсов [IApcConnection](#), [IApcSession](#), [IApcServerWrap](#) и [IApcObjectWrap](#).

Класс	Описание
<i>TApcConnection</i>	Экземпляры этого класса позволяют установить соединение с сервером APACS Bio. Класс реализует интерфейс <a href="#">IApcConnection</a> .
<i>TApcSession</i>	Экземпляры этого класса предоставляют сеанс связи с сервером APACS Bio и реализуют интерфейс <a href="#">IApcSession</a> .

**Проверка адресов регистрации COM-объектов**

Неверные адреса регистрации COM-объектов могут служить причиной отказа в работе SDK APACS Bio.

Проверить адреса регистрации COM-объектов можно, запустив утилиту **«Адреса COM-объектов»**.

В случае, когда адреса регистрации COM-объектов отличаются от фактических адресов, следует перерегистрировать COM-объекты, запустив файл APACS Bio\bin\ApcRegSDKDlls.bat. Перерегистрация объектов произойдет автоматически.

Проверить правильность перерегистрации можно нажав кнопку **Обновить** в окне утилиты **«Адреса СОМ-объектов»**, при этом адреса регистрации должны поменяться на верные.

1.4.1 Интерфейс **IApcConnection**

Позволяет установить новое соединение с сервером APACS Bio.

Метод	Описание
<a href="#">createSession</a>	Метод создает новое соединение с сервером APACS Bio.

1.4.1.1 Метод **createSession**

Метод создает новое соединение с сервером APACS Bio.

C++

```
HRESULT createSession ( BSTR astrLoginName/*[in]*/,
                        BSTR astrPasswd/*[in]*/,
                        IApcSession** apSession/*[out]*/,
                        int* apnResult/*[out,retval]*/)
```

**Параметры**

[in] astrLoginName — имя оператора в системе APACS Bio,  
[in] astrPasswd — пароль оператора в системе APACS Bio,  
[out] apSession — сессия соединения с сервером APACS Bio ([IApcSession](#)),  
[out] apnResult — 0 — если соединение с сервером установлено, иначе — код ошибки.

**Возвращаемое значение**

Возвращает значение HRESULT.

**Рекомендации**

Смотрите пример установления соединения с сервером APACS Bio.

```
// COM
CoInitializeEx( NULL, COINIT_APARTMENTTHREADED ) ;
//
CComPtr<IApcConnection> spCon;
HRESULT hRes = spCon.CoCreateInstance( CLSID_TApcConnection,
    NULL,
    CLSCTX_INPROC ) ;
if( hRes == S_OK )
{
    //
    CComPtr<IApcSession> spSes;
    int nResult = -1;
    CComBSTR login( L"Inst" ) ;
    CComBSTR pswrd( L"" ) ;
    hRes = spCon->createSession( login,
        pswrd,
        &spSes,
        &nResult ) ;
    if( hRes == S_OK && nResult == 0 )
    {
        // APACS Bio
        CComPtr<IApcServerWrap> spApacs;
        CComPtr<IDispatch> spDispApacs;
        hRes = spSes->getServer( &spDispApacs ) ;
    }
}
```

```
if( hRes == S_OK)
{
    hRes = spDispApacs.QueryInterface( &spApacs) ;
    if( hRes == S_OK)
    {
        //                APACS Bio
    }
}

//
spSes->close( ) ;
}
```

1.4.1.2 Метод createSession2

Метод создает новое соединение с сервером APACS Bio.

C++

```
HRESULT createSession2 (BSTR astrLoginName/*[in]*/,
                        BSTR astrPasswd/*[in]*/,
                        Apcsrvsdk_tlb::IApcSession** apSession/
*[out,retval]*/)
```

Параметры

[in] astrLoginName — имя оператора в системе APACS Bio,  
[in] astrPasswd — пароль оператора в системе APACS Bio,  
[out] apSession — сессия соединения с сервером APACS Bio ([IApcSession](#)),  
[out] apnResult — 0 — если соединение с сервером установлено, иначе — код ошибки.

Возвращаемое значение

Возвращает значение HRESULT.

1.4.2 Интерфейс IApcSession

Сессия соединения с сервером APACS Bio.

Для доступа к объектной модели сервера APACS Bio используйте метод [getServer](#).

Для завершения сеанса работы с сервером APACS Bio — метод [close](#).

Для получения уведомления об обрыве соединения с сервером APACS Bio — метод [set\\_onDisconnect](#).

Смотрите пример установления соединения с сервером APACS Bio [>>](#)

Метод	Описание
<a href="#">close</a>	Метод завершает сеанс связи с сервером APACS Bio.
<a href="#">getServer</a>	Метод позволяет получить объект, поддерживающий интерфейс <a href="#">IApcServerWrap</a> .
<a href="#">set_onDisconnect</a>	Метод позволяет установить обработчик на обрыв с соединения с сервером APACS Bio.

### 1.4.2.1 Метод `getServer`

Метод позволяет получить объект, поддерживающий интерфейс [IApcServerWrap](#).

#### C++

```
HRESULT getServer( LPDISPATCH* appServer/*[out,retval]*/) 
```

#### Параметры

[out] `appServer` – указатель на объект, поддерживающий интерфейс [IApcServerWrap](#).

#### Возвращаемое значение

Возвращает значение `HRESULT`.

### 1.4.2.2 Метод `close`

Метод завершает сеанс связи с сервером APACS Bio.

#### C++

```
HRESULT close( void) 
```

#### Возвращаемое значение

Возвращает значение `HRESULT`.

### 1.4.2.3 Метод `set_onDisconnect`

Метод позволяет установить обработчик на обрыв с соединения с сервером APACS Bio. При обрыве соединения будет вызван метод `Invoke` с `DispID=0`, без параметров.

#### C++

```
HRESULT STDMETHODCALLTYPE set_onDisconnect( LPDISPATCH apDisHandler/*[in]*/) ,
```

#### Параметры

[in] `apDisHandler` — указатель на объект, который будет принимать уведомления о разрыве соединения с сервером.

#### Возвращаемое значение

Возвращает значение `HRESULT`.