

Возможности подключения к серверу (для программиста 1С)



Внимание! Подключаемая компонента поддерживает только 32-битные версии платформы 1С.

В данном руководстве представлены описания функций и свойств компоненты **AddIn_1CToWebServer.dll**, которая используется для подключения 1С Предприятие 8.2(3) к ПО «Сервер ВЕСЫ АВТО».



Обратите внимание, в папка Examples которая идет вместе с дистрибутивом, выложены примеры конфигураций для подключения к ПО Сервер ВЕСЫ АВТО.

Основные функции:

ОткрытьСерверВесы (<АдресСервера>)

Подключиться к Серверу Весов

<АдресСервера> – сетевой путь (IP адрес) к компьютеру, где установлен драйвер. Если подключение локально, то пустая строка «Возвращает 1-если успешно, 0-неудачно»



Для 1С версии 8.x удаленное подключение не работает, только локально <АдресСервера> = ««

Для удаленного подключения используйте компоненту **AddIn_1CToWebServer.dll** и дополнительное ПО "**WEB Сервер ВЕСЫ АВТО**"

Пример подключения Add1CServerVesy.dll из макета:

Попытка

```
глВесы = Неопределено;  
ПодключитьВнешнююКомпоненту("ОбщийМакет.ВнешняяКомпонентаСерверАвто1", "Server  
Auto", ТипВнешнейКомпоненты.Native);  
глВесы = Новый("AddIn.ServerAuto.Add1CServerVesy");  
глВесы.Включен = 1;
```

```
глВесы.ОткрытьСерверВесы(Сервер);
    Возврат Истина;
Исключение
    Сообщить("ВНИМАНИЕ!! Ошибка вызова
глВесы1.ОткрытьСерверВесы(""+Сервер+"");
    глВесы = Неопределено;
    Возврат Ложь;
КонецПопытки;
```

ЗакретьСерверВесы()

Процедура закрытия сервера весов. Запускать можно несколько раз подряд.

СерверОткрыт

Функция возвращает 1, если сервер весов успешно открыт.

ВключитьСобытия()

Включить генерирование внешнего события **UpdateMassa**. для обработки событий сервера используется процедура **ОбработкаВнешнегоСобытия(Источник,Событие,Данные)**; источник **UpdateMassa**.

ВключитьСобытия()

Включить генерирование внешнего события **UpdateMassa**. для обработки событий сервера используется процедура **ОбработкаВнешнегоСобытия(Источник,Событие,Данные)**; источник **Add1CServerVesy**. По умолчанию включено.

Пример обработчика:

```
Процедура ВнешнееСобытие(Источник, Событие, Данные)
Если (Источник = "Add1CServerVesy") Тогда
    Если Событие = "UpdateMassa" Тогда

        ИначеЕсли Событие = "UpdatefSTABIL" Тогда

        ИначеЕсли Событие = "UpdateFreq" Тогда

        ИначеЕсли Событие = "UpdateStateVesyName" Тогда

        ИначеЕсли Событие = " UpdateCamRecognTS" Тогда

        ИначеЕсли Событие = "UpdateCamNumbTS" Тогда

        ИначеЕсли Событие = "UpdateCamNumbPricep" Тогда

        ИначеЕсли Событие = "UpdateAutoVsv" Тогда

        ИначеЕсли Событие = "UpdateShit1" Тогда

        ИначеЕсли Событие = "UpdateShit2" Тогда

        ИначеЕсли Событие = " UpdateCam1RecognTS" Тогда
```

```
ИначеЕсли Событие = " UpdateCam2RecognTS " Тогда
ИначеЕсли Событие = " UpdateCam3RecognTS " Тогда
ИначеЕсли Событие = " UpdateCam4RecognTS " Тогда

КонецЕсли ;
КонецПроцедуры
```

ОтключитьСобытия()

Отключить генерирование внешнего события **Update...**

Масса

Функция возвращает значение массы.

Частота

Функция возвращает частоту пакетов, посылаемых весовым индикатором в компьютер.

УстановитьНоль()

Послать команду установки нуля на весовой индикатор.

Ноль

Функция возвращает флаг «Ноль» (значение 1 или 0).

Стабил *вызывается перед взвешиванием, для проверки*

Функция возвращает флаг «Стабильность» (значение 1 или 0).

ИмяВесов

Функция возвращает имя весов.

ЭмуляцияПоказаний

Функция возвращает 1 если включен режим эмуляции.

ВходПользователя (<ИмяПользователя>)

Регистрирует вход пользователя в базу событий Сервера Весы <ИмяПользователя> – Строка.
Имя пользователя, подключившегося к весам.

ВыходПользователя (<ИмяПользователя>)

Регистрирует выход пользователя в базу событий Сервера Весы <ИмяПользователя> – Строка.
Имя пользователя, подключившегося к весам.

ИмяСостоянияВесов

Возвращает строку сокращенного наименования состояния весов

РасшифровкаСостоянияВесов

Возвращает строку расшифровки состояния весов

Функции контроля взвешивания:

ЗафиксироватьВзвешивание(<GUIDДокумента>, <МАССА>, <ИмяПользователя>);



ОБЯЗАТЕЛЬНО ДЛЯ ИСПОЛЬЗОВАНИЯ

Устанавливает флаг контроля, и сообщает Серверу о том, что взвешивание произведено.

Следующая за ней команда **ЗаписатьДокументВзвешивания**.

GUIDДокумента – уникальный идентификатор взвешивания. Если пуст, то заполняется новым значением.

МАССА – масса взвешенного груза.

ИмяПользователя – Строка. Имя пользователя.

ПослеВзвешиванияНеСъезжал() вызывается перед взвешиванием, для проверки съехал ли транспорт с весов.

Функция возвращает флаг “ПослеВзвешиванияНеСъезжал” (значение 1 или 0)

ВзвешиваниеНеСНоля() вызывается перед взвешиванием, для проверки было ли обнуление весов перед заездом транспорта.

Функция возвращает флаг “ВзвешиваниеНеСНоля ” (значение 1 или 0)

Функции выгрузки документа взвешивания в дублирующий журнал Сервера Весы:

ЗаписатьДокументВзвешивания(<GUIDДокумента>, <КодТипаВзвешивания>, <ТипВзвешивания>, <НомерДокумента>, <ДатаДокумента>, <ВремяДокумента>, <КодОтправитель>, <Отправитель>, <КодПолучатель>, <Получатель>, <КодПунктОтправления>, <ПунктОтправления>, <КодПунктНазначения>, <ПунктНазначения>, <КодГруз>, <Груз>, <КодВодитель>, <Водитель>, <НомерНакладной>, <ДатаНакладной>, <ТараПоДокументу>, <БруттоПоДокументу>, <НеттоПоДокументу>, <КодМаркаТС>, <МаркаТС>, <НомерТС,РегионТС>, <НомерПрицепа>, <РегионПрицепа>, <ИмяПользователя>, ТАРА>, ДатаТарирования>, <ВремяТарирования>, <БРУТТО>, <ДатаБрутирования>, <ВремяБрутирования>, <НЕТТО>, <флагУдален>)

Создает или обновляет в базе взвешиваний запись о взвешивании с заполненными полями. Если GUID (уникальный идентификатор взвешивания) не найден создает новую запись, иначе обновляет существующую **GUIDДокумента** – уникальный идентификатор взвешивания, тип строка GUID. Если <GUIDДокумента> пустой, создается при первой записи сервером Firebird.

ТипВзвешивания – Строки - наименование типа взвешивания.

НомерДокумента – Номер документа.

ДатаДокумента – дата документа.

ВремяДокумента – время документа.

КодОтправителя, Отправитель – Код и наименование Контрагенты

КодПолучатель, Получатель – Код и наименование Контрагенты

КодПунктОтправления, ПунктОтправления – Код и наименование Пункты или МестаХранения

Код ПунктНазначения, ПунктНазначения – Код и наименование Пункты или МестаХранения.

Код Груз, Груз – Код и наименование Номенклатура.

КодВодитель, Водитель – Код и ФИО водителя.

НомерНакладной – строка НомерНакладной.

ДатаНакладной – Дата накладной.

ТараПоДокументу – тара по документу.

БруттоПоДокументу – брутто по документу.

НеттоПоДокументу – нетто по документу.

Код МаркаТС, МаркаТС – Код и наименование Марки ТС.

НомерТС,РегионТС – Гос.номер и регион ТС.

НомерПрицепа – Гос. Номер прицепа

РегионПрицепа – регион прицепа.

ИмяПользователя – Строка. Имя пользователя.

ТипТары – Строка. Характер значения тары- “взвешивание”или “вручную”.

ТАРА, ДатаТарирования,ВремяТарирования – Строка масса тары, дата и время тарирования

ТипБрутто – Строка. Характер значения брутто- “взвешивание”или “вручную”.

БРУТТО, ДатаБрутирования,ВремяБрутирования – Строка масса брутто, дата и время брутир.

НЕТТО – Масса НЕТТО.

флагУдален – признак флага удален ('1' – удален, '0' – не удален).



поля <Код..> используются в базе взвешиваний для ведения справочников.

ЗаписатьПараметрДокументаВзвешивания(<GUIDДокумента>, <ИмяПоля>,<Значение>)

Создает или обновляет в базе взвешиваний запись о взвешивании Обновляет значение указанного поля.

Если GUID (уникальный идентификатор взвешивания) не найден создает новую запись, иначе обновляет существующую

GUIDДокумента – уникальный идентификатор взвешивания, тип строка GUID. Если <GUIDДокумента> пустой, создается при первой записи сервером Firebird.

ЗаписатьСобытие(<ТипСобытия>,<Описание>,<Пользователь>)

Создает в фискальном журнале событий запись о дополнительном событии. Регистрация текущим временем.

<**ТипСобытия**>- метка (целое число) указывающая на тип события

41 - Внешнее событие

<**Описание**> – Строка. Описание события

<**Пользователь**>- Строка. Имя пользователя

Функции работы с видеокameraми:

JPG *полном размере*

ПолучитьФотоКамеры1();

JPG *в размере окна*

ПолучитьФотоПоРазмеруКамеры1(<Ширина>,<Высота>);

БМП *в полном размере*

ПолучитьФотоБМПКамеры1();

БМП в размере окна

ПолучитьФотоБМППоРазмеруКамеры1(<Ширина>,<Высота>);

Пример Сохранения фото в файл:

```
БинДанные = ТекущиеВесы.ПолучитьФотоКамеры1();  
Фото1 = Новый Картинка(БинДанные);  
Фото1.Записать(КаталогИБ+ "\PHOTO\"+GUIDДокументаВзвешивания +  
"_PHOTO_"+НомерВзв+"1.jpg");
```

Функции системы распознавания номеров:

Флаг номер найден

КамераНайденНомер свойство возвращает - 1, если номер найден

Получить НомерТС

КамераНомерТС возвращает строку с номером

Получить РегионТС

КамераРегионТС возвращает строку с регионом

Получить НомерПрицепа

КамераНомерПрицеп возвращает строку с номером

Получить РегионПрицепа

КамераРегионПрицеп возвращает строку с регионом

Функции оборудования УДВВ и контроля проезда транспорта:

РасшифровкаСостоянияПроезда

Строка расшифровка текущего состояния проезда

СобытиеНомерВКамере1

Возвращает 0 или 1. Событие - стабильное распознавание номера в камере 1

СобытиеНомерВКамере2

Возвращает 0 или 1. Событие - стабильное распознавание номера в камере 2

СобытиеНомерВКамере3

Возвращает 0 или 1. Событие - стабильное распознавание номера в камере 3

СобытиеНомерВКамере4

Возвращает 0 или 1. Событие - стабильное распознавание номера в камере 4

СобытиеДатчик1

Возвращает 0 или 1. Событие - сработал датчик 1

СобытиеДатчик2

Возвращает 0 или 1. Событие – сработал датчик 2

РазрешитьПроезд

Возвращает 0 или 1. Флаг –проезд разрешен

РазрешитьПроезд1

Возвращает 0 или 1. Флаг –проезд разрешен в направлении 1

РазрешитьПроезд2

Возвращает 0 или 1. Флаг –проезд разрешен в направлении 2

УДВВ_Частота

Возвращает частоту обмена с устройством ввода-вывода

УДВВ_СостояниеВыходы

Состояние выходов устройства ввода-вывода

УДВВ_СостояниеВходы

Состояние входов устройства ввода-вывода

УДВВ_УстановитьВыход(int NBit, int _Value)

Команда установки значения выхода, нужного бита

УстановитьРазрешитьПроезд()

Команда установить флаг проезд разрешен

УстановитьРазрешитьПроезд1()

Команда установить флаг проезд разрешен в направлении 1

УстановитьРазрешитьПроезд2()

Команда установить флаг проезд разрешен в направлении 2

Функции загрузки автоматических взвешиваний, создаваемых Сервером ВЕСЫ Авто:

Сервер Весы Авто при переходе в автоматический режим создает записи взвешиваний независимо от 1С. Данный набор функций позволяют последовательно скопировать данные автовзвешивания начиная с первой с помощью последовательности команд:

1. Анализ количество новых автовзвешиваний:

```
Если ГлВесы.КоличествоНовыхАвтоВзвешиваний > 0 Тогда
```

2. Выбрать первое

```
ГлВесы.ВыбратьАвтоВзвешивание ( )
```

3. Дождаться когда будет выбрано

```
Если ГлВесы.АвтовзвешиваниеВыбрано = 1 Тогда
```

4. Читать параметры выбранного автовзвешивания в новый документ:

```
GUIDДокументаВзвешивания = ГлВесы.ПолучитьПараметрАвтоВзвешивания ("CODE");
    ДатаДок = ГлВесы.ПолучитьПараметрАвтоВзвешивания ("DATE");
ВремяДок = ГлВесы.ПолучитьПараметрАвтоВзвешивания ("TIME");
    ***** И Т.Д.
```

5. Записать новое взвешивание:

```
ГлВесы.ЗаписатьДокументВзвешивания (...
```

После этого сбрасывается **ГлВесы.АвтовзвешиваниеВыбрано = 0**, далее по новой.

флаг состояние режима автовзвешивания

РежимАвтоВзвешивания – возвращает 0 или 1

команда установить режим автовзвешивания

УстановитьРежимАвтоВзвешивания(Режим) <Режим> – 0 или 1

количество незагруженных автовзвешиваний

КоличествоНовыхАвтоВзвешиваний – возвращает целое число от 0 и больше.

Команда выбора первого автовзвешивания

ВыбратьАвтоВзвешивание() – результат не возвращает

флаг готовности для загрузки выбранного автовзвешивания

АвтовзвешиваниеВыбрано – возвращает 0 или 1

получить параметр загруженного автовзвешивания

ПолучитьПараметрАвтоВзвешивания(ИмяПараметра) <ИмяПараметра> – 0 или 1

Получить фото первого отвеса загруженного автовзвешивания

ПолучитьФОТОАвтоВзвешОтвес1(НомерФото) <НомерФото> – номер камеры 1, 2, 3 или 4

Получить фото второго отвеса загруженного автовзвешивания

ПолучитьФОТОАвтоВзвешОтвес2(НомерФото) <НомерФото> – номер камеры 1, 2, 3 или 4

```
//-----
---
// Алгоритм загрузки новых взвешиваний из Сервера Весы
//-----
---
Процедура ОбработкаОжиданияВК()
//-----
    Если КоличествоНовыхАвтоВзвешиваний1 <> глВесы.КоличествоНовыхАвтоВзвешиваний
Тогда
    КоличествоНовыхАвтоВзвешиваний1 = глВесы.КоличествоНовыхАвтоВзвешиваний;
        Если (глВесы1.АвтовзвешиваниеВыбрано = 0) и
(КоличествоНовыхАвтоВзвешиваний1>0) Тогда
            глВесы.ВыбратьАвтоВзвешивание();
            АвтовзвешиваниеВыбрано1 = -1;
        КонецЕсли;
```



```
КонецЕсли;  
//-----  
Если АвтовзвешиваниеВыбрано1 <> глВесы1.АвтовзвешиваниеВыбрано Тогда  
АвтовзвешиваниеВыбрано1 = глВесы1.АвтовзвешиваниеВыбрано;  
Если (АвтовзвешиваниеВыбрано1 = 1) и (КоличествоНовыхАвтоВзвешиваний1>0)  
Тогда  
СоздатьАвтоВзвешивание(); // процедура чтения параметров и создания записи о  
взвешивании и сохранения  
КоличествоНовыхАвтоВзвешиваний1 = -1;  
КонецЕсли;  
КонецЕсли;  
  
ПодключитьОбработчикОжидания("ОбработкаОжиданияВК", 0.2, Истина);  
КонецПроцедуры
```

From:

<https://docuwiki.vesysoft.ru/> - База знаний

Permanent link:

<https://docuwiki.vesysoft.ru/doku.php?id=manager:1c>

Last update: **2021/05/25 11:28**

