

Глава 8. Прочие функции

Ход вычислений	2
СгенерироватьGUID.....	2
ЗагрузитьФункции.....	3
ВыполнитьВыражение.....	3
Отладить.....	4
ВерсияПрограммы.....	5
ТекущаяЗадача.....	6
ПерейтиВЗадачу.....	6
Внешний обработчик	7
СледФункция.....	7
Экспорт/импорт.....	8
ЭкспортироватьЗапись.....	8
ЭкспортироватьВыборку.....	9
ИмпортироватьФайл.....	10
ЗаписатьКарточкуVCard.....	11
ПрочитатьКарточкуVCard.....	13
ОсновнаяТаблица.....	14
ВнешнееПоле.....	14
ЕстьВнешнееПоле.....	15
ЗаписьНайдена.....	16
КодировкаФайла.....	17
ФорматФайла.....	17
Конфигурация системы.....	18
ЗаписатьКонфигурацию.....	18
ПрочитатьКонфигурацию.....	19
Учетная запись.....	19
УчетнаяЗапись.....	19
ПослатьПисьмо.....	20
ПослатьПисьмоДокумент.....	21
Очистить.....	22
ПолучитьПочту.....	23
Формирование штрих-кода.....	24
ШтрихКодДокумента.....	24
ШтрихКодДокументаКонтрольный.....	25
ПроверитьШтрихКодДокументаКонтрольный.....	25
НадоПечататьКонтрольныйШтрихКод.....	26
EAN8.....	27
EAN13.....	27
I25.....	28
Диаграмма СБиС++.....	29
НайтиОбъект.....	29
Параметры.....	30
УстановитьПараметры.....	31
Воспроизведение звука.....	32
ПроигратьФайл.....	32
Решение технических проблем.....	33
СохранитьМинидамп.....	33
Функции ЭО.....	33

Отправитель.....	33
Инспектор	34
Есть Действительный Сертификат	35
Проверить Незавершенные Запросы На Сертификат	35
Проверить Новые Сертификаты Для Запросов	36

Ход вычислений

Сгенерировать GUID

Генерирует уникальный код.

Синтаксис

Сгенерировать GUID ()

Параметры

Не указываются.

Возвращает

Текст - сгенерированный уникальный код.

Комментарии

GUID (Globally Unique Identifier) — представляет собой статистически уникальный 128-битный идентификатор. Его главная особенность — уникальность, которая позволяет создавать расширяемые сервисы и приложения без опасения получения конфликтов вызванных совпадением идентификатора. Хотя уникальность каждого отдельного GUID не гарантируется, общее количество уникальных ключей настолько велико (2^{128} или $3,4028 \times 10^{38}$), что вероятность того, что один и тот же ключ будет получен дважды — очень мала.

GUID — это частная реализация (фирмой Microsoft) стандарта, имеющего название Universally Unique Identifier (UUID).

В тексте GUID записывается в виде строки из 16-ричных цифр, разбитых на группы дефисами и окружённой фигурными скобками:

```
{6F9619FF-8B86-D011-B42D-00CF4FC964FF}
```

Пример

```
перем пGUID = СгенерироватьGUID(); # в переменной  
пGUID, например, "24eedf1a-ba05-4b6a-82b2-  
28d5480101f2"
```

ЗагрузитьФункции

Загружает функции из файла.

Синтаксис

```
ЗагрузитьФункции (ИмяФайла1, [ИмяФайла2] ...)
```

Параметры

ИмяФайла1, **ИмяФайла2** (текст) – имя файла, который необходимо открыть. Файл, содержащий функции имеет расширение «.prg», его (расширение) указывать не надо.

Возвращает

Да (логическое значение) – если удалось загрузить функции.

Нет (логическое значение) – ничего не было загружено.

Примеры

```
пКол = ЗагрузитьФункции ("ЭлектроннаяОтчетность"); #  
в пКол «Да»;
```

ВыполнитьВыражение

Выполняет указанное выражение.

Синтаксис

```
ВыполнитьВыражение (Выражение)
```

Параметры

Выражение (текст или двоичные данные) – выражение на внутреннем языке, которое необходимо выполнить.

Возвращает

Значение любого типа - результат выполнения выражения.



Пример

```
оФайл = Файл("test.exe");
оФайл.Кодировка("dos");
Пока(!оФайл.ЭтоКонец())
{
    оФайл.ПрочитатьСтроку(пСтрока);
    ВыполнитьВыражение(пСтрока);
}; # из файла "test.exe" построчно считываются и
выполняются строки написанные на внутреннем языке.

оФ = Файл(«Proba2.txt»);
Пока(!оФ.ЭтоКонец())
{
    оФ.ПрочитатьСтроку(пСтрока);
    ВыполнитьВыражение(«Сообщить(пСтрока)»);
}; # из файла «Proba2.txt» построчно считываются и
сообщаются строки.
```

Отладить

Выводит иерархический список всех доступных в данный момент переменных, объектов и полей.

Синтаксис

```
Отладить([ИмяПеременной])
```

Параметры

ИмяПеременной (переменная) – переменная, доступная в данный момент. Если указан параметр у функции, то при выводе иерархического списка будет выделена курсором указанная переменная. Если этой переменной нет, то функция покажет список всех доступных в данный момент переменных, объектов и полей.

Комментарий

Функцию удобно использовать, если вы не знаете, какие данные доступны в определённый момент выполнения правила, или не знаете имени переменной, содержащей искомое значение. Функция также может пригодиться, когда в процессе отладки операции нужно просмотреть значения переменных.



В верхней строке окна выводится имя текущей переменной. Комбинацией <Ctrl+Ins> можно скопировать полное имя текущей переменной с тем, чтобы потом комбинацией <Shift+Ins> вставить имя переменной в текст операции.

Перед именами объектов и полей связи выводится желтый прямоугольник. На таких строках можно нажать <Enter> и просмотреть элементы объекта или поля связанной записи.

В локальном меню окна также есть команды «Прервать выполнение» (завершает выполнение операции или отчёта) и «Выполнить выражение» (выражение выполняется в текущем контексте, и сразу выводится результат).

Возвращает

Ничего не возвращает.

Примеры

Необходимо посмотреть данные по нашей организации:

```
оЛицо = Лицо0 ( ) ;  
Отладить ( ) ;
```

Чтобы сразу увидеть название нашей организации, надо написать:

```
оЛицо = Лицо0 ( ) ;  
Отладить ( оЛицо . Наименование ) ;
```

ВерсияПрограммы

Возвращает строку с номером версии текущей программы, состоящую из цифр, разделенных точкой.

Синтаксис

```
ВерсияПрограммы ( )
```

Параметры

Не указываются.

Возвращает

Текст с номером версии программы.



Примеры

Поскольку функция появилась только в версии 1.9, то для проверки более поздних версий можно использовать следующий фрагмент:

```
Если (Есть (ВерсияПрограммы) )  
    Сообщить (ВерсияПрограммы ( ) ) ;  
Иначе  
    Сообщить ("1.8 и ниже") ;
```

ТекущаяЗадача

Возвращает название текущей задачи программы.

Синтаксис

```
ТекущаяЗадача ( )
```

Параметры

Не указываются.

Возвращает

Текст – название задачи, в которой работаем на данный момент.

Примеры

```
ТекущаяЗадача ( ) ; # возвращает название «Товары и  
материалы», если в данный момент работаем с данными  
этого раздела программы.
```

См. также:

- [ПерейтиВЗадачу](#)

ПерейтиВЗадачу

Переходит в раздел указанной задачи программы (аналогично нажатию кнопки F11).

Синтаксис

```
ПерейтиВЗадачу (ИмяЗадачи)
```



Параметры

ИмяЗадачи (текст) – название задачи, в раздел которой необходимо перейти. Необходимо точно указывать название задачи, соблюдая регистры. В случае, если неправильно указано название задачи, либо задачи нет в программе, функция предложит воспользоваться окном выбора необходимой задачи.

Комментарии

Если нужно воспользоваться окном выбора задачи, то можно воспользоваться функцией ВыполнитьКоманду("TaskSwitch").

Возвращает

Да (логическое значение) – открыта указанная задача.

Нет (логическое значение) – не открыта указанная задача.

Примеры

ПерейтиВЗадачу ("Электронная отчетность"); #
переходит в раздел задачи «Электронная отчетность».

См. также:

- [ТекущаяЗадача](#)

Внешний обработчик

СледФункция

Запускает выполнение следующего обработчика.

Синтаксис

СледФункция ()

Параметры

Не указываются.

Возвращает

Любой.

Комментарии

Список внешних обработчиков можно посмотреть нажав <Ctrl+Shift+F12> на таблице.

Экспорт/импорт

ЭкспортироватьЗапись

Экспортирует запись выборки из программы во внешний файл, аналогично команде экспорта данных, вызываемой при нажатии комбинации клавиш <Ctrl+U>.

Синтаксис

ЭкспортироватьЗапись (**ЗаписьВыборки**, **ИмяКаталога** [, **ИмяКонфигурации**])

Параметры

ЗаписьВыборки (объект) – запись указанной выборки, которую нужно экспортировать во внешний файл.

ИмяКаталога (текст) – название каталога, в котором находится импортируемый внешний файл (файл с расширением, указанном в формате экспорта/импорта).

ИмяКонфигурации (текст) – название конфигурации экспорта – импорта. Если не указана, используется конфигурация по умолчанию.

Возвращает

Да (логическое значение) – данные импортированы.

Нет (логическое значение) – данные не импортированы.

Примеры

Экспортируем запись сотрудника в указанный каталог:

```
oВыборка = Выборка ("Сотрудники") ;
oВыборка.ТабНомер = "106" ;
Найти ("Сотрудники", "ТабНомер") ;
ЭкспортироватьЗапись (oВыборка, "d:\\_Work\\2\\") ;
```

См. также:

- [Запись](#)
- [Выборка](#)
- [ЭкспортироватьВыборку](#)
- [ИмпортироватьФайл](#)

ЭкспортироватьВыборку

Экспортирует записи выборки из программы во внешний файл, аналогично команде экспорта данных, вызываемой при нажатии комбинации клавиш <Ctrl+U>.

Синтаксис

ЭкспортироватьВыборку (**Выборка** , **ИмяКаталога** [, **ИмяКонфигурации** [, **ДействиеСФайлами**]])

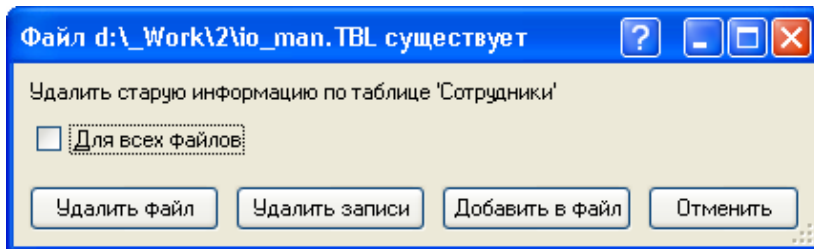
Параметры

Выборка (объект) – выборка, записи которой нужно экспортировать во внешний файл.

ИмяКаталога (текст) – название каталога, в котором находится импортируемый внешний файл (файл с расширением, указанном в формате экспорта/импорта).

ИмяКонфигурации (текст) – название конфигурации экспорта – импорта. Если не указана, используется конфигурация по умолчанию. Необходимо указывать точное название конфигурации со всеми пробелами и чёрточками, а также соблюдая регистры.

ДействиеСФайлами (текст) – указывается действие, которое должно выполниться, если при экспорте будут обнаружены существующие файлы. Параметр может принимать следующие значения: "**добавить**" - добавить записи в существующий файл; "**очистить**" – удалить записи из существующего файла; "**удалить**" - удалить файл. Если параметр не указан, то при существовании файлов будет возникать запрос:



Обратите внимание на то, что при указании данного параметра, обязательно должно быть указано либо конкретное **ИмяКонфигурации**, либо пустая строка "".

Возвращает

Да (логическое значение) – данные импортированы.

Нет (логическое значение) – данные не импортированы.

Примеры

Экспортируем все записи справочника во внешний файл:

```
oВыборка = Выборка ("Сотрудники");
ЭкспортироватьВыборку (oВыборка, "d:\\_work\\2\\");
```

Добавим во внешний файл новые записи справочника:

```
oВыборка = Выборка ("Сотрудники");
ЭкспортироватьВыборку (oВыборка, "d:\\_work\\2\\", "-
По умолчанию -", "добавить");
```

См. также:

- [Выборка](#)
- [ЭкспортироватьЗапись](#)
- [ИмпортироватьФайл](#)

ИмпортироватьФайл

Импортирует записи из внешнего файла в программу, аналогично команде импорта данных, вызываемой при нажатии комбинации клавиш <Ctrl+I>.

Синтаксис

ИмпортироватьФайл (**ЗаписьВыборки**, **ИмяКаталога** [, **ИмяКонфигурации**])

ИмпортироватьФайл (**ИмяВыборки**, **ИмяКаталога** [, **ИмяКонфигурации**])

Параметры

ЗаписьВыборки (объект) – запись выборки, импортируемой из внешнего файла в программу.

ИмяВыборки (текст) – имя выборки, в которую импортируются записи.

ИмяКаталога (текст) – название каталога, в котором находится импортируемый внешний файл (файл с расширением, указанным в формате экспорта/импорта).

ИмяКонфигурации (текст) – название конфигурации экспорта – импорта. Если не указана, используется конфигурация по умолчанию.

Возвращает

Да (логическое значение) – данные импортированы.

Нет (логическое значение) – данные не импортированы.

Примеры

ИмпортироватьФайл ("Сотрудники", "d:_Work\\2\\"); # импортируем данные из внешнего файла в справочник сотрудников.

См. также:

- [ЭкспортироватьЗапись](#)
- [ЭкспортироватьВыборку](#)

ЗаписатьКарточкуVCard

Экспортирует записи выборки в файлы VCard (в файлы с расширением «.vcf»).

Синтаксис

ЗаписатьКарточкуVCard (**Имяфайла**, **Названиеформата**, **Запись1** [, **Запись2**], ...)

Параметры

ИмяФайла (текст) – путь к файлу, в который необходимо сохранить все результаты. Необходимо указывать полный путь к файлу, иначе функция ищет этот файл в каталоге программы СБиС++.

НазваниеФормата (текст) - название формата экспорта для данного справочника. Список доступных форматов указан в коллекции записей «Форматы для работы с VCard» в файле **СБиС.rs**.

Запись1, Запись2,... (объект) - записи из выборки базы данных. Эти записи обязательно должны быть однотипными, т.е. из одной выборки.

Возвращает

Да (логическое значение) – если созданы файлы VCard.

Нет (логическое значение) – если создать файлы VCard не удалось.

Примеры

Необходимо экспортировать запись сотрудника с табельным номером «1»:

```
oВыб=Выборка ("Сотрудники" );
Если (Найти (oВыб, "ТабНомер", 1)
Сообщить (ЗаписатьКарточкуVCard ("C:\\", "Общий
минимальный формат", oВыб) );
```

Необходимо экспортировать несколько записей сотрудников:

```
перем pНом=0;
ДляВсех (Записей ("Сотрудники" )
{
    Если (pНом==5)
        Прервать ;
    Иначе
    {
        ЗаписатьКарточкуVCard ("Карточки", "Общий
минимальный формат", Запись);
        pНом++;
    }
}
```

См. также:

- [ПрочитатьКарточкуVCard](#)

ПрочитатьКарточкуVCard

Импортирует записи в выборку из файла VCard (из файла с расширением «.vcf»).

Синтаксис

ПрочитатьКарточкуVCard (**Выборка** , **НазваниеФормата** , **ИмяФайла**)

Параметры

Выборка (объект) - объект выборки, в которую помещается создаваемая запись.

Каталог (текст) – путь к файлу с расширением **.vcf**, из которого необходимо импортировать записи в выборку. Необходимо указывать полный путь к файлу, иначе функция ищет этот файл в каталоге программы СБиС++.

НазваниеФормата (текст) - название формата экспорта для данного справочника. Список доступных форматов указан в коллекции записей «Форматы для работы с VCard» в файле **СБиС.rs**.

Возвращает

Да (логическое значение) – если на основании импортируемых данных созданы записи в справочнике.

Нет (логическое значение) – если записи в справочнике по какой-то причине не созданы.

Примеры

Необходимо импортировать записи в выборку из файла:

```
oВыборка = Выборка ("Сотрудники" );  
Сообщить (ПрочитатьКарточкуVCard (oВыборка , "Формат  
Outlook" , "C:\\VCard.vcf" ) ;
```

См. также:

- [ЗаписатьКарточкуVCard](#)

ОсновнаяТаблица

Проверяет тип таблицы (или выборки) при выполнении экспорта/импорта записей.

Синтаксис

ОсновнаяТаблица ()

Параметры

Не указываются.

Возвращает

Да (логическое значение) – если экспортируется / импортируется основная таблица.

Нет (логическое значение) – если экспортируется / импортируется связанная таблица.

Примеры

Экспортируем записи справочника «Сотрудники» во внешний файл. При этом будут экспортированы и записи из связанных таблиц, например, «Банки». В результате функция возвратит различные значения:

```
функция ПослеЭкспортаЗаписи ( )  
{  
    пОснТаб = ОсновнаяТаблица ( ) ;  
    Сообщить (пОснТаб) ;  
}
```

ВнешнееПоле

Возвращает имя внешнего поля файла, соответствующего внутреннему полю таблицы (или выборки) в описании формата.

Синтаксис

ВнешнееПоле (ВнутрПоле)

Параметры

ВнутрПоле (текст) – название внутреннего поля.



Возвращает

Текст – имя внешнего поля.

Примеры

Экспортируем записи справочника «Сотрудники». При этом, если в описании формата полю выборки «Адрес» указано соответствующее поле файла, то функция возвратит название этого поля:

```
функция ПослеЭкспортаЗаписи ()
{
    пВнешПоле = ВнешнееПоле ("Адрес") ;
    Сообщить (пВнешПоле) ;
}
```

См. также:

- [ЕстьВнешнееПоле](#)

ЕстьВнешнееПоле

Проверяет существование внешнего поля, соответствующего указанному внутреннему полю в описании формата.

Синтаксис

ЕстьВнешнееПоле (ВнутрПоле)

Параметры

ВнутрПоле (текст) – название внутреннего поля.

Возвращает

Да (логическое значение) – если есть внешнее поле, соответствующее внутреннему.

Нет (логическое значение) – если нет внешнего поля, соответствующего внутреннему.

Примеры

После экспорта записей справочника «Сотрудники» функция возвратит «Да» если есть внешнее поле соответствующее полю «Адрес», иначе - «Нет»:



```
функция ПослеЭкспортаЗаписи (
{
    пЕстьВнешПоле = ЕстьВнешнееПоле ("Адрес" ) ;
    Сообщить (пЕстьВнешПоле) ;
}
```

См. также:

- [ВнешнееПоле](#)

ЗаписьНайдена

Проверяет существование записи по значению ключевого поля, указанному в описании формата в поле «Ключевой индекс».

Синтаксис

ЗаписьНайдена ()

Параметры

Не указываются.

Возвращает

Да (логическое значение) – запись найдена.

Нет (логическое значение) – запись не найдена.

Комментарии

Действие функции имеет смысл только при импорте записей.

Примеры

При выполнении импорта записей из внешнего файла в справочник «Сотрудники», функция возвратит «Да», если данная запись найдена в справочнике:

```
функция ПослеИмпортаЗаписи (
{
    пЗапНайдена = ЗаписьНайдена ( ) ;
    Сообщить (пЗапНайдена) ;
}
```

КодировкаФайла

Возвращает название кодировки внешнего файла, указанного в поле «Кодировка» в описании формата.

Синтаксис

КодировкаФайла ()

Параметры

Не указываются.

Возвращает

Текст – название кодировки. Возможные значения: DOS, Windows, как в конфигурации.

Примеры

При экспорте записей справочника «Сотрудники» возвратит название кодировки внешнего файла:

```
функция ПослеЭкспортаЗаписи ( )
{
    пКодировка = КодировкаФайла ( ) ;
    Сообщить (пКодировка) ;
}
```

См. также:

- [ФорматФайла](#)

ФорматФайла

Возвращает название формата внешнего файла, указанного в поле «Кодировка» в описании формата.

Синтаксис

ФорматФайла ()

Параметры

Не указываются.

Возвращает

Текст – название формата внешнего файла, например: DBF, TBL и т.д.

Примеры

При экспорте записей справочника «Сотрудники» возвратит название формата внешнего файла:

```
функция ПослеЭкспортаЗаписи ()
{
    пФормат = ФорматФайла ();
    Сообщить (пФормат) ;
}
```

См. также:

- [КодировкаФайла](#)

Конфигурация системы

ЗаписатьКонфигурацию

Сохраняет данные конфигурации в файле sbis.cfg.

Синтаксис

ЗаписатьКонфигурацию (Имя , НовОбъект)

Параметры

Имя (текст) – произвольное имя конфигурации.

НовОбъект (объект) – объект, который надо записать в файл.

Возвращает

Число - размер записанных данных в файл в байтах.

Примеры

```
перем пД[] ;
пД.Имя = «Сидоров»;
пД.В = ТекВремя;
пД.Д = ТекДата;
ЗаписатьКонфигурацию («Моя заметка», пД) ; # данные
```



конфигурации «Моя заметка» сохраняются в файле `sbis.cfg`.

См. также:

- [ПрочитатьКонфигурацию](#)

ПрочитатьКонфигурацию

Читает данные конфигурации из файла `sbis.cfg`.

Синтаксис

`ПрочитатьКонфигурацию (Имя)`

Параметры

Имя (текст) – имя конфигурации.

Возвращает

Объект, содержащий данные конфигурации.

Примеры

```
oK = ПрочитатьКонфигурацию («Моя заметка»); #  
считывает данные из конфигурации Моя заметка.  
oK =  
ПрочитатьКонфигурацию («ЭлектроннаяОтчетность»); #  
считывает данные из Электронной Отчетности.
```

См. также:

- [ЗаписатьКонфигурацию](#)

Учетная запись

УчетнаяЗапись

Возвращает реквизиты указанной учетной записи. Этот объект содержит настройки почтового ящика (имя учетной записи, электронная почта, параметры настройки сервера и т.д.), необходимого для обработки сообщений в процессе работы с электронной почтой в задаче «Документооборот».



Синтаксис

`УчетнаяЗапись (Строка)`

Параметры

Строка (текст) – имя существующей учетной записи, указанной в настройках почтового ящика в соответствующем поле.

Возвращает

Объект - объект с реквизитами указанной учетной записи. Если учетной записи не существует, программа выдаст об этом соответствующее сообщение.

Примеры

Узнаем адрес электронной почты учетной записи «Сидоров»:

```
oУчетЗапись = УчетнаяЗапись ("Сидоров") ;  
ЕслиЕсть (oУчетЗапись)  
    Сообщить (oУчетЗапись . ЭлПочта) ;
```

См. также:

- [ПослатьПисьмо](#)
- [ПослатьПисьмоДокумент](#)
- [Очистить](#)
- [ПолучитьПочту](#)

ПослатьПисьмо

Отсылает письмо с почтового ящика отправителя по указанному адресу с прикрепленными файлами, если есть.

Синтаксис

```
УчЗапись . ПослатьПисьмо (Адрес [ , Тема [ , Текст [ ,  
Имяфайла1 | файл1 ] [ , Имяфайла2 | файл2 ] . . . )
```

Параметры

Адрес (текст) – адрес электронной почты получателя.



Тема (текст) – тема письма.

Текст (текст или двоичные данные) – текст отправляемого письма.

ИмяФайла1|Файл1, ИмяФайла2|Файл2,... (текст|объект) – строка с указанием полного пути к вложенному файлу или объект, созданный при помощи функции «Файл». В качестве параметра можно указать каталог, тогда при отправке к письму будут прикреплены все файлы, которые есть в этом каталоге.

Возвращает

Да (логическое значение) – если письмо отправлено.

Нет (логическое значение) – если письмо не отправлено (например, не удалось установить соединение с почтовым сервером).

Примеры

Отправим письмо с вложением по адресу «test@tensor.ru»:

```
oУчетЗапись = УчетнаяЗапись ("Сидоров") ;  
oУчетЗапись.ПослатьПисьмо ("test@tensor.ru",  
"Ознакомление", "Ознакомься с документом, указанным  
во вложении", "C:\\Мои документы\\файл.doc") .
```

См. также:

- [УчетнаяЗапись](#)
- [ПослатьПисьмоДокумент](#)
- [Очистить](#)
- [ПолучитьПочту](#)

ПослатьПисьмоДокумент

Отсылает письмо с почтового ящика отправителя по указанному адресу с прикрепленными файлами, и создает документ в реестре документов электронной почты.

Синтаксис

```
УчЗапись.ПослатьПисьмоДокумент (Адрес [, Тема [,  
Текст [, Имяфайла1|файл1] [, Имяфайла2|файл2] ...])
```



Параметры

Адрес (текст) – адрес электронной почты получателя.

Тема (текст) – тема письма.

Текст (текст или двоичные данные) – текст отправляемого письма.

ИмяФайла1|Файл1, ИмяФайла2|Файл2,... (текст|объект) – строка с указанием полного пути к вложенному файлу или объект, созданный при помощи функции «Файл». В качестве параметра можно указать каталог, тогда при отправке к письму будут прикреплены все файлы, которые есть в этом каталоге.

Возвращает

Объект – запись документа из реестра документов электронной почты.

См. также:

- [УчетнаяЗапись](#)
- [ПослатьПисьмо](#)
- [Очистить](#)
- [ПолучитьПочту](#)

Очистить

Удаляет письма с почтового ящика данной учетной записи до указанной даты.

Синтаксис

УчЗапись . Очистить (Дата)

Параметры

Дата (дата) – все письма данной учетной записи, даты которых меньше указанной, будут удалены.

Возвращает

Да (логическое значение) – если письма удалены.



Нет (логическое значение) – если не удалось удалить письма.

Примеры

Удалим старые письма с почтового ящика «Сидорова», оставим только письма с текущей датой:

```
oУчетЗапись = УчетнаяЗапись ("Сидоров") ;  
oУчетЗапись .Очистить (ТекДата) ;
```

См. также:

- [УчетнаяЗапись](#)
- [ПослатьПисьмо](#)
- [ПослатьПисьмоДокумент](#)
- [ПолучитьПочту](#)

ПолучитьПочту

Запускает процесс получения писем для данной учетной записи.

Синтаксис

```
УчЗапись .ПолучитьПочту ( )
```

Параметры

Не указываются.

Комментарии

Весь сеанс получения почты отслеживается в журнале сообщений, который открывается сразу после вызова функции.

Возвращает

Да (логическое значение) – если почта успешно получена.

Нет (логическое значение) – если не удалось получить почту (например, не удалось установить соединение с почтовым сервером).

Примеры

Необходимо получить почту для учетной записи «Сидоров»:

```
oУчетЗапись = УчетнаяЗапись ("Сидоров") ;
```

`oУчетЗапись . ПослатьПисьмо () .`

См. также:

- [УчетнаяЗапись](#)
- [ПослатьПисьмо](#)
- [ПослатьПисьмоДокумент](#)
- [Очистить](#)

Формирование штрих-кода

ШтрихКодДокумента

Позволяет формировать штрих-код документа в формате готовом для печати при помощи шрифта SbisI25.

Синтаксис

`ШтрихКодДокумента (Документ)`

Параметры

`Документ` (объект) – документ, для которого формируется штрих-код.

Возвращает

`Текст` – строка символов.

Примеры

Сформируем штрих-код для указанного документа:

```
oДокумент = Таблица ("Документы" );  
oНовЗапись = oДокумент . Запись ( ) ;  
oНовЗапись . Загрузить (2315) ;  
Сообщить (ШтрихКодДокумента (oДокумент) ) ;
```

См. также:

- [I25](#)
- [ШтрихКодДокументаКонтрольный](#)



ШтрихКодДокументаКонтрольный

Позволяет формировать штрих-код контрольной суммы документа в формате готовом для печати при помощи шрифта SbisI25.

Синтаксис

ШтрихКодДокументаКонтрольный (Документ)

Параметры

Документ (объект) – документ, для которого формируется штрих-код.

Возвращает

Текст – строка штрих-кода (строка символов) для печати шрифтом SbisI25 (шрифт контрольной суммы).

Примеры

Сформируем обычный и контрольный штрих-коды для указанного документа:

```
оДокумент = Таблица ("Документы" );
оНовЗапись = оДокумент . Запись ( ) ;
оНовЗапись . Загрузить ( 2315 ) ;
Сообщить ( ШтрихКодДокумента ( оДокумент ) ) ;
Сообщить ( ШтрихКодДокументаКонтрольный ( оДокумент ) ) ;
```

См. также:

- [ШтрихКодДокумента](#)

ПроверитьШтрихКодДокументаКонтрольный

Проверяет, соответствует ли переданная контрольная сумма контрольной сумме указанного документа.

Синтаксис

**ПроверитьШтрихКодДокументаКонтрольный (Документ ,
КонтрСумма)**

Параметры

Документ (объект) – документ, контрольная сумма которого сравнивается с передаваемой контрольной суммой.



КонтрСумма (текст) – цифровая строка контрольной суммы.

Возвращает

Да (логическое значение) – контрольная сумма верна.

Нет (логическое значение) – контрольная сумма неверна.

См. также:

- [ШтрихКодДокумента](#)
- [ШтрихКодДокументаКонтрольный](#)
- [НадоПечататьКонтрольныйШтрихКод](#)

НадоПечататьКонтрольныйШтрихКод

Осуществляет контроль за выводом на печать контрольного штрих-кода в складских документах.

Синтаксис

НадоПечататьКонтрольныйШтрихКод ()

Параметры

Без параметров.

Комментарии

В печатных формах документа штрих-код остаётся справа, а слева будет штрих-код с контрольной суммой. Действие функции аналогично установке флага в поле «**Печатать контрольный штрих-код**» в окне настройки конфигурации задачи «Торговля и логистика».

Возвращает

Да (логическое значение) – необходимо печатать штрих-код контрольной суммы документа (если установлен флаг в поле «**Печатать контрольный штрих-код**»).



Нет (логическое значение) – не надо печатать штрих-код контрольной суммы документа (флаг в поле «**Печатать контрольный штрих-код**» не установлен).

См. также:

- [ШтрихКодДокумента](#)
- [ШтрихКодДокументаКонтрольный](#)
- [ПроверитьШтрихКодДокументаКонтрольный](#)

EAN8

Формирует специальную строку с символами для вывода на печать из штрих-кодов в формате EAN-8 с помощью шрифта SbisEAN13.

Синтаксис

EAN8 (Строка)

Параметры

Строка (текст) – строка, состоящая из цифр. Для EAN8 строка должна содержать 7 символов (функция сама рассчитает и добавит необходимую контрольную цифру).

Возвращает

Текст – строка с символами.

Примеры

EAN8 ("4767889") ;

См. также:

- [EAN8](#)

EAN13

Формирует специальную строку с символами для вывода на печать из штрих-кодов в формате EAN-13 с помощью шрифта SbisEAN13.

Синтаксис



EAN13 (Строка)

Параметры

Строка (текст) – строка, состоящая из цифр. Для EAN13 строка должна содержать 12 символов (функция сама рассчитает и добавит необходимую контрольную цифру).

Возвращает

Текст – строка с символами.

Примеры

Сообщить (EAN13 ("476122123123")) ;

См. также:

- [EAN13](#)

I25

Формирует специальную строку с символами для вывода на печать из штрих-кодов с помощью шрифта Sbis I25.

Синтаксис

I25 (Строка)

Параметры

Строка (текст) – строка, состоящая из цифр. Для I25 количество символов в строке ограничено возможностями сканеров. Контрольной суммы не существует, если число цифр в коде нечетное, то в начале будет добавлен "0".

Возвращает

Текст – строка с символами.

Примеры

```
<script language = sbis>  
Перем пКод;  
пКод = I25 ("123"); # в пКод находится значение
```

«(1G)»

```
Напечатать ( "<td style=\"font-family:SBIS i25; font-size:35;\"><field> пКод </field></td> " );  
</script>; # на экране будет выведен штрих-код с символами 123.
```

См. также:

- [ШтрихКодДокумента](#)

Диаграмма СБиС++

НайтиОбъект

Находит запись диаграммы или легенды в указанном диалоге.

Синтаксис

`Диалог.НайтиОбъект(НазваниеДиаграммыИлиЛегенды)`

Параметры

НазваниеДиаграммыИлиЛегенды (текст) – название диаграммы СБиС++ или легенды, привязанной к указанному диалогу. Добавление такого объекта выполняется при создании или редактировании (<Ctrl+F12>) диалога из группы «Оформление» («Создать»).

Возвращает

Объект, запись диаграммы или легенды в диалоге. Если такого объекта нет, то функция возвращает пустой объект.

Пример

Использование данной функции приводится в примере к функции «Параметры».

См. также:

- [Параметры](#)
- [УстановитьПараметры](#)

Параметры

Возвращает установленные параметры диаграммы или легенды.

Синтаксис

`Объект . Параметры ()`

Параметры

Не указываются.

Комментарии

Полный список параметров диаграммы или легенды представлен в ресурсе **СБиС.rs** (формат записи «Свойства диаграммы» или «Свойства легенды»). Возвращаемый объект содержит массив «РядыДанных» с заполненными полями ввода. Каждое поле ввода в свою очередь представляет собой объект с установленными значениями типа «Текст» и «Цвет». Дополнительно возвращаемый объект может содержать выборку, привязанную к диаграмме.

Для извлечения записи диаграммы или легенды необходимо использовать функцию «**НайтиОбъект**».

Возвращает

Объект, содержащий полный перечень свойств и их значений, установленных для данной диаграммы или легенды.

Пример

Узнаем установленные параметры диаграммы «График» в диалоге «Пример»:

```
oДиалог=Диалог ("Пример") ;  
oОбъект=oДиалог . НайтиОбъект ("График") ;  
oПараметры=oОбъект . Параметры ( ) ;  
Отладить (oПараметры)
```

См. также:

- [НайтиОбъект](#)
- [УстановитьПараметры](#)

УстановитьПараметры

Устанавливает параметры для диаграммы или легенды в диалоге.

Синтаксис

Объект . УстановитьПараметры (Параметры)

Параметры

Параметры (объект) – запись формата «Свойства диаграммы» или «Свойства легенды», содержащий список возможных параметров, которые необходимо установить или изменить для диаграммы или легенды.

Комментарии

Полный список параметров диаграммы представлен в ресурсе **СБиС.rs** (формат записи «Свойства диаграммы» или «Свойства легенды»). Для получения списка возможных параметров в диаграмме или легенде необходимо сначала использовать функцию «Параметры», а уже потом устанавливать новые значения этих параметров.

Возвращает

Да (логическое значение) – значения параметров установлены.

Нет (логическое значение) – не удалось установить новые значения параметров.

Пример

Изменим фон и подписи по оси абсцисс у диаграммы «График» в диалоге «Пример»:

```
oДиалог=Диалог ("Пример") ;
oОбъект=oДиалог . НайтиОбъект ("График") ;
oПараметры=oОбъект . Параметры ( ) ;
OPАРАМЕТРЫ . ПодписьОсиАбсцисс="ФИО сотрудников" ;
OPАРАМЕТРЫ . ПрозрачныйФон=Да ;
Сообщить (oОбъект . УстановитьПараметры (oПараметры) ) ;
```

См. также:

- [Параметры](#)
- [НайтиОбъект](#)



Воспроизведение звука

ПроигратьФайл

Воспроизводит файл с расширением **.wav** на аудиоустройство по умолчанию.

Синтаксис

ПроигратьФайл (ПутьКФайлу)

Параметры

ПутьКФайлу (текст) – полный путь к файлу, который необходимо воспроизвести. Иначе функция ищет файл в каталоге программы СБиС++.

Комментарии

Функция находится во внутреннем обработчике "Воспроизведение звука". Отработка функции сопровождается звуковым сигналом, согласно указанному файлу.

Возвращает

Ничего не возвращает.

Пример

Пусть есть диалог «Пример», на который повешен обработчик «Воспроизведение звука». В этом диалоге есть кнопка, при нажатии на которую будет вызываться данная функция – СБиС++ воспроизведет звук, переданный в качестве параметра этой функции:

```
оДиалог=Диалог ("Пример" ) ;  
оДиалог.Выполнить ;
```

Решение технических проблем

СохранитьМинидамп

Сохраняет дампы памяти на момент вызова функции. Файл с дампом помещается в каталог "dumps" рядом с файлом sbis.exe.

Синтаксис

СохранитьМинидамп ()

Параметры

Не указываются.

Комментарии

Функция используется для выявления ошибок и отладки программы. Файл с дампом формируется в формате "dump_<Дата>_<Время>.dmp". В дальнейшем созданные файлы можно отправить разработчикам для оперативного решения возникшей проблемы.

Возвращает

Да (логическое значение) – если файл с дампом сохранен.

Нет (логическое значение) – в процессе сохранения возникли ошибки.

Функции ЭО

Отправитель

Возвращает текущего выбранного налогоплательщика, который в данный момент является отправителем отчетности.

Синтаксис

Отправитель ()

Параметры

Не указываются.



Возвращает

Объект – запись раздела документов текущего выбранного отправителя из таблицы «Документы», от которого можно брать **ЛицоП**. Если налогоплательщик не выбран, то функция вернет пустую запись.

Примеры

Узнаем текущего отправителя отчетности:

```
oОтпр=Отправитель ( ) ;
```

```
Если (oОтпр)
```

```
Сообщить (oОтпр . ЛицоП ( ) . Название) ;
```

Инспектор

Возвращает инспектора – сотрудника, закрепленного за указанной организацией.

Синтаксис

```
Инспектор (Организация)
```

Параметры

Организация (объект) – запись организации из таблицы «Лица», для которой нужно узнать закрепленного инспектора.

Комментарии

В программе определение инспектора – сотрудника, закрепленного за конкретной организацией, выполняется на папке организаций в поле «Инспектор».

Возвращает

Объект – запись лица из таблицы «Сотрудники». Функция ищет такую запись по иерархии.

Примеры

Узнаем ФИО сотрудника, прикрепленного к текущей организации:

```
oЛицо=Лицо ("7605016030", "Орг") ;
```

```
oИнспектор=Инспектор (oЛицо) ;
```

Сообщить (ОИнспектор.ФИО) ;

Есть Действительный Сертификат

Проверяет наличие у лица действительного сертификата.

Синтаксис

Есть Действительный Сертификат (Лицо)

Параметры

Лицо (объект) – запись лица из таблицы «Лица», для которой нужно проверить наличие действительного сертификата.

Комментарии

Список всех сертификатов организации отображается в карточке этой организации на закладке «Сертификаты». У действительного сертификата в этом списке будет статус «Действителен». По этому статуса функция и определяет действующий сертификат.

Возвращает

Да (логическое значение) – найден действительный сертификат.

Нет (логическое значение) – действительный сертификат не найден.

Примеры

Узнаем наличие действительного сертификата у данной организации:

`ОЛицо=Лицо ("7605016030", "Орг") ;`

`Сообщить (Есть Действительный Сертификат (ОЛицо)) ;`

Проверить Незавершенные Запросы На Сертификат

Проверяет наличие незавершенных запросов на сертификат в БД и, в случае их наличия, предлагает проверить ответы на них.

Синтаксис

Проверить Незавершенные Запросы На Сертификат ()

Параметры

Не указываются.



Комментарии

За создание запроса отвечает кнопка «Создать запрос на сертификат» в карточке налогоплательщика. В окне запроса отображаются фазы обработки запроса и их состояния. Функция проверяет наличие в списке незавершенных запросов, если таковые есть, то предлагает проверить ответы на них.

Возвращает

Ничего не возвращает.

См. также:

- [ПроверитьНовыеСертификатыДляЗапросов](#)

ПроверитьНовыеСертификатыДляЗапросов

В случае наличия в БД незавершенных запросов на сертификат проверяет съемные диски на наличие файлов-сертификатов соответствующих данным запросам на сертификат и, в случае наличия соответствующего сертификата, предлагает установить его.

Синтаксис

`ПроверитьНовыеСертификатыДляЗапросов ()`

Параметры

Не указываются.

Комментарии

Возвращает

Ничего не возвращает.

См. также:

- [ПроверитьНезавершенныеЗапросыНаСертификат](#)