Тип «BarcodeData» в Mobile SMARTS

Последние изменения: 2024-03-26

BarcodeData — тип, который используется для получения информации о штрихкоде/сканировании.

С помощью данного типа можно проверить совместимость штрихкода с форматом GS1, получить набор идентификаторов применения по штрихкоду, а также, при сканировании камерой, получить изображение штрихкода.

Свойства

Свойство
Описание
string BarcodeCleared
Очищенный от различных контрольных символов RAW штрихкод
string BarcodeGS1Cleared
Очищенный от различных контрольных символов GS1 штрихкод
string BarcodeGS1Formatted
Отформатированный GS1 штрихкод в формате «со скобками»
string BarcodeGS1Printable
Корректный правильный GS1 штрихкод, пригодный для печати, с символами-разделителями групп
IPicture BarcodePicture
Полное изображение отсканированного ШК*
IPicture BarcodePictureWithBorder
Изображение отсканированного ШК с рамкой вокруг ШК*
string BarcodeRaw
Исходный поток данных ШК, с контрольными символами и т. д.
string BarcodeRawEx
Исходный поток данных ШК, с контрольными символами и т. д. Еще более сырой, чем BarcodeRaw*
string BeerShortMark
Код идентификации пивной и слабоалкогольной продукции по стандартам Честного знака
string BeerShortMarkFormatted
Код идентификации пивной и слабоалкогольной продукции в формате «со скобками»

string BikeShortMark
Код идентификации товарной группы «Велосипеды» по стандартам Честного знака
string BikeShortMarkFormatted
Код идентификации товарной группы «Велосипеды» в формате «со скобками»

IPicture CropedBarcodePicture
Изображение ШК, обрезанное по рамке вокруг ШК*
string DrugsShortMark
Код идентификации товарной группы «Лекарства» по стандартам Честного знака
string DrugsShortMarkFormatted
Код идентификации товарной группы «Лекарства» в формате «со скобками»
string EAN13
EAN13 из отсканированного ШК
string EAN8
EAN8 из отсканированного ШК
string ErrorGS1Compatible
Ошибка после проверки совместимости ШК с форматом GS1
string FashionShortMark
Код идентификации товаров легкой промышленности по стандартам Честного знака
string FashionShortMarkFormatted
Код идентификации товаров легкой промышленности в формате «со скобками»
Ean128 GS1
Объект GS1, содержащий подробную информацию об используемых идентификаторах применения
string GTIN
GTIN в том виде, в котором он может быть получен из ШК
string GTINBase
GTIN с лидирующим символом 0
bool Hardware
Означает, был ли создан объект сканированием или программно
bool IsAlcoMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки алкогольной продукции
bool IsBeerMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки пивной и слабоалкогольной продукции по стандартам Честного знака
bool IsBikeMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки товарной группы «Велосипеды» по стандартам Честного знака
bool IsDrugsMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки товарной группы «Лекарства» по стандартам Честного знака
bool IsFashionMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки товаров легкой промышленности по стандартам Честного знака

bool IsGS1Compatible
Удалось ли разложить ШК на группы идентификаторов применения по стандарту GS1
bool IsGTINCompatible
Удалось ли найти GTIN в отсканированном ШК
bool IsMark
Проверяет совместимость отсканированного штрихкода с каждым из форматов кода маркировки
bool IsMilkMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки молочной продукции по стандартам Честного знака
bool IsPerfumeryMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки духов и туалетной воды по стандартам Честного знака
bool IsPhotoMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки фотоаппаратов и ламп-вспышек по стандартам Честного знака
bool IsShoesMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки товарной группы «Обувь» по стандартам Честного знака
bool IsTiresMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки шин по стандартам Честного знака.
bool IsTobaccoBox
Проверяет совместимость отсканированного ШК с форматом кода маркировки блока табачной продукции по стандартам Честного знака
bool IsTobaccoMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки блока или пачки табачной продукции по стандартам Честного знака
bool IsTobaccoPack
Проверяет совместимость отсканированного ШК с форматом кода маркировки пачки табачной продукции по стандартам Честного знака
bool IsWaterMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки товарной группы «Упакованная вода» по стандартам Честного знака
bool IsWheelChairMark
Проверяет совместимость отсканированного ШК с форматом кода маркировки товарной группы «Кресла-коляски» по стандартам Честного знака
string MilkShortMark
Код идентификации молочной продукции по стандартам Честного знака
string MilkShortMarkFormatted
Код идентификации молочной продукции в формате «со скобками»

string PerfumeryShortMark
Код идентификации духов и туалетной воды по стандартам Честного знака

string PerfumeryShortMarkFormatted
Код идентификации духов и туалетной воды в формате «со скобками»
string PhotoShortMark
Код идентификации фотоаппаратов и ламп-вспышек по стандартам Честного знака
string PhotoShortMarkFormatted
Код идентификации фотоаппаратов и ламп-вспышек в формате «со скобками»
string ScannedBarcodeCompatible
Возвращает штрихкод в том виде, в котором он раньше всегда попадал в переменную сессии ScanedBarcode
string ShoesShortMark
Код идентификации товарной группы «Обувь» по стандартам Честного знака
string ShoesShortMarkFormatted
Код идентификации товарной группы «Обувь» в формате «со скобками»
string ShortMark
Код идентификации любой из групп по стандартам Честного знака
string ShortMarkFormatted
Код идентификации любой из групп в формате «со скобками»
string StrBarcodeType
Тип штрихкода, распознанный сканером. Если сканер не распознал или не умеет этого, то UNKNOWN_TYPE.
string TiresShortMark
Код идентификации шин по стандартам Честного знака
string TiresShortMarkFormatted
Код идентификации шин в формате «со скобками»
string TobaccoBoxShortMark
Код идентификации блока табачной продукции по стандартам Честного знака
string TobaccoBoxShortMarkFormatted
Код идентификации блока табачной продукции в формате «со скобками»
string TobaccoPackShortMark
Код идентификации пачки табачной продукции по стандартам Честного знака
string TobaccoPackShortMarkFormatted
Код идентификации пачки табачной продукции в формате «со скобками»
string TobaccoShortMark
Код идентификации блока или пачки табачной продукции по стандартам Честного знака
string TobaccoShortMarkFormatted
Код идентификации блока или пачки табачной продукции в формате «со скобками»

string UPCA
Штрихкод UPC-A, если он таковым является
string WaterShortMark
Код идентификации товарной группы «Упакованная вода» по стандартам Честного знака
string WaterShortMarkFormatted
Код идентификации товарной группы «Упакованная вода» в формате «со скобками»
string WheelChairShortMark
Код идентификации товарной группы «Кресла-коляски» по стандартам Честного знака
string WheelChairShortMarkFormatted
Код идентификации товарной группы «Кресла-коляски» в формате «со скобками»
string WiFiPass
Пароль для подключения к Wi-Fi **
string WiFiSSID
Название сети для подключения к Wi-Fi **
string WiFiType
Тип шифрования для подключения к Wi-Fi **

* при сканировании камерой устройства на ОС Android
** при сканировании устройством на ОС Android

Всю основную информацию о штрихкоде можно получить с помощью свойств данного типа, однако, если содержимое сформировано и успешно распознано с использованием идентификаторов применения GS1 может возникнуть потребность обращаться к данным через коды идентификаторов применения. Это возможно сделать с помощью методов свойства GS1.

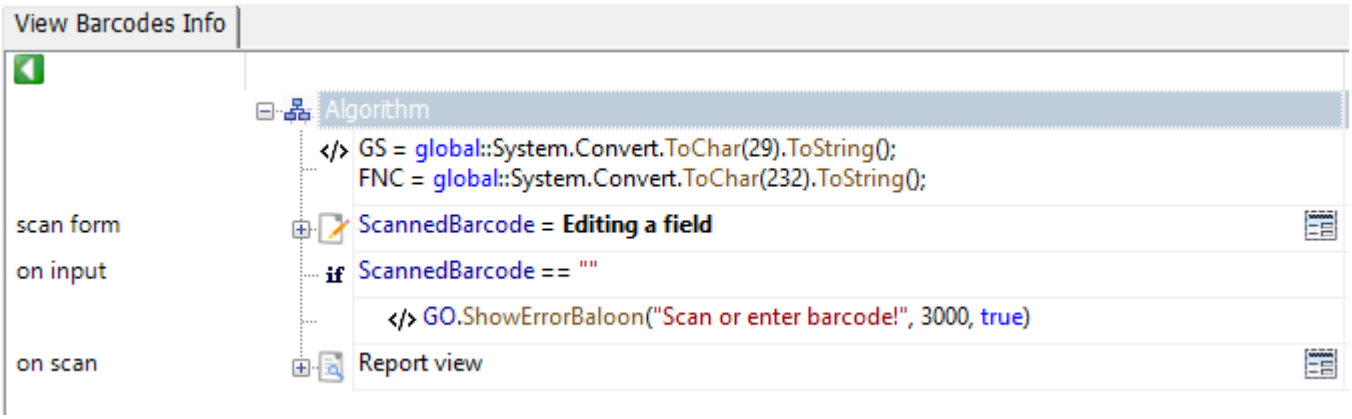
Методы свойства GS1

Метод
Описание
bool Contains (ai)
Проверяет наличие идентификатора применения среди групп разобранного ШК
IEnumerable GetEntries ()
Получает перечисление групп идентификаторов применения вместе со значениями
object GetObjectValue (ai)
Возвращает значение по переданному идентификатору применения из разобранного ШК
string GetStringValue (ai)
Возвращает строковое значение по переданному идентификатору применения из разобранного ШК
string GetValue (ai)
Возвращает строковое значение по переданному идентификатору применения из разобранного ШК

Примеры использования

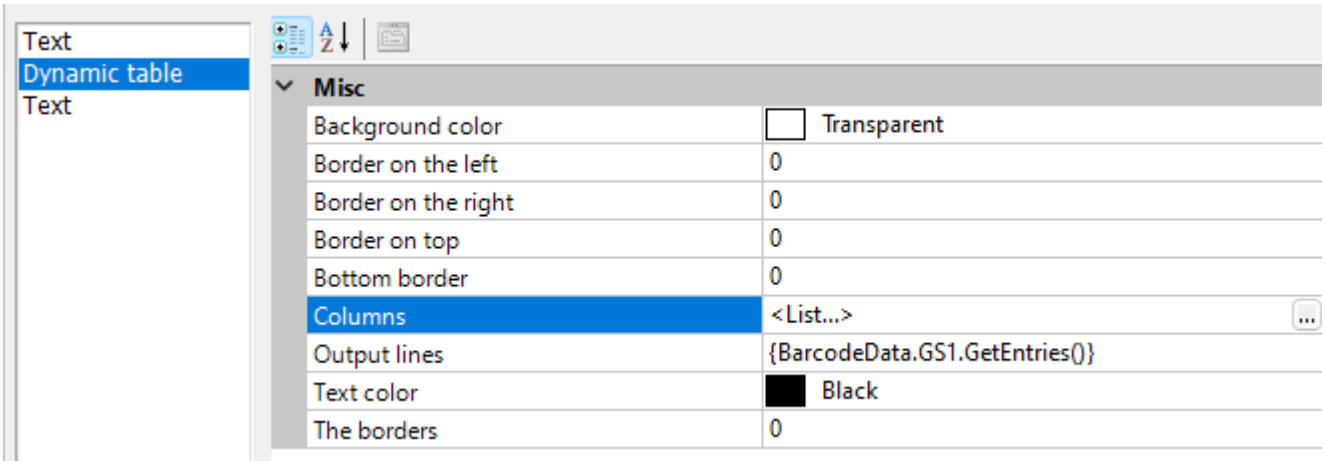
Задача: Получить всю доступную информацию по данным штрихкода.

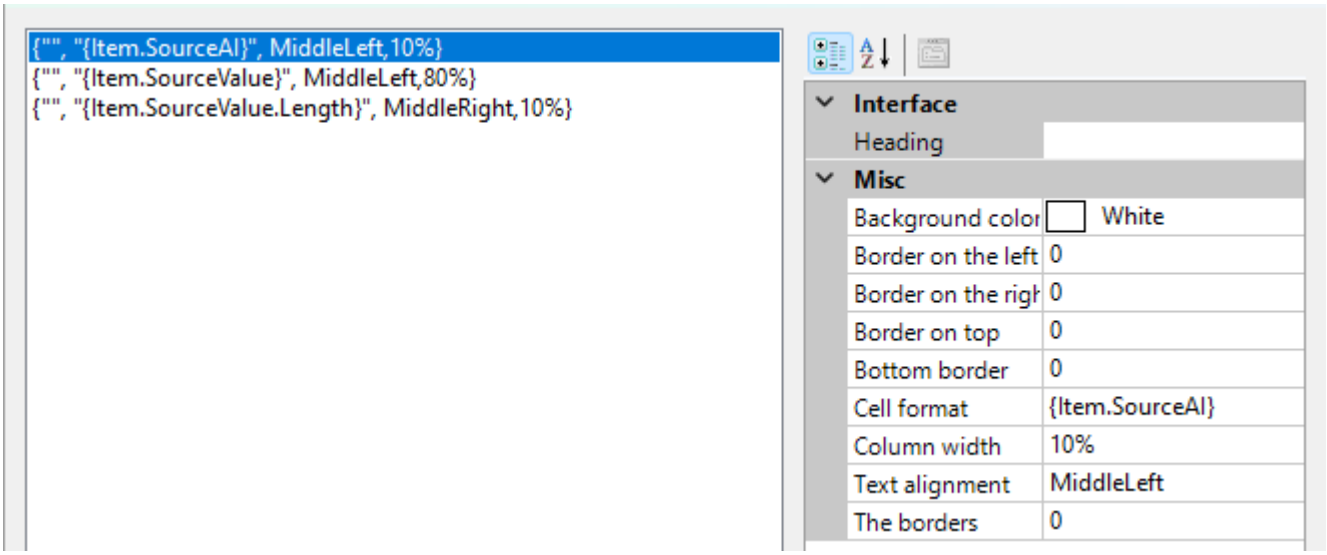
Решение:




```
Report viev.Report blocks[0]
   <div>
   <p class="__helper_text" style="padding-top: 8dp;">Barcode type</p>
   <p>{BarcodeData.StrBarcodeType}</p>
   <p class="__helper_text" style="padding-top: 8dp;">Raw barcode</p>
   <p>{BarcodeData.BarcodeRawEx.Replace (GS, «<blue>{CDATA («<GS>»)}</blue>»).Replace
(FNC, «<red>{CDATA («<FNC1>»)}</red>»)}</p>
   {BarcodeData.IsGS1Compatible:<div>
   <p class="__helper_text" style="padding-top: 8dp;">Human-readable interpretation</p>
   <p>{BarcodeData.BarcodeGS1Formatted}</p>
   </div>;}
   {BarcodeData.IsMark:<div>
   <p class="__helper_text" style="padding-top: 8dp;">Track & Trace identification code</p>
   <p>{BarcodeData.ShortMarkFormatted}</p>
   </div>;}
   {BarcodeData.IsGS1Compatible:<div>
   <p class="__helper_text" style="padding-top: 8dp;">Application identifiers</p>
   </div>;}
   </div>
```

```
Report viev.Report blocks[1]
```






```
Report viev.Report blocks[2]
  <div style="padding-bottom: 100dp;">
  {BarcodeData.CropedBarcodePicture != null:<div>
  <p class="__helper_text» style="padding-top: 8dp;">Picture</p>
  <img size="stretch">{BarcodeData.CropedBarcodePicture}</img>
  </div>;}
  </div>
```

Рассмотрим пример подробнее:

- **В первом блоке отчета выводим на экран**
- **Тип отсканированного штрихкода**
- **Сырые данные штрихкода в виде «как есть»**
- **Приведенные к человекочитаемому формату данные***
- **Код идентификации для системы прослеживаемости товаров***
- **Заголовок для списка идентификаторов применения***
  *** для случаев, когда такая возможность есть**
- **Во втором блоке выводим список идентификаторов**

- В третьем блоке выводим изображение отсканированного ШК*
  * при сканировании камерой

Код примера

Список типов кодировок

AustraliaPost
EAN8
PosiCode
Aztec
GRIDMATRIX
Postals
BooklandEAN
GS1_128
Postnet
BPO
GS1_DATABAR
QR
CanadaPost
GS1_DATABAR_Expanded
RSS
ChinaPost
GS1_DATABAR_Limited
RSS14
Codabar
GS1_DATABAR_OMNIDIRECTIONAL
RSSExpanded
Codablock
GS1_Databar14
RSSLimited
Code11
GS1_DATAMATRIX
Straight2of5
Code128
GS1_QR
Straight2of5_IATA

Code128Emultion
HANXIN
Straight2of5_Industrial

Code16k
HK25
Telepen
Code32
IDTag
TLC39
Code32_PARAF
Interleaved2of5
TriopticCode39
Code39
ISBT128
UKPostal
Code49
JapanPost
UNKNOWN_TYPE
Code93
KoreanPost
UPCA
Composite_CCAB
Matrix2of5
UPCA_2Supps
Composite_CCC
MaxiCode
UPCA_5Supps
Composite_TLC39
MESA
UPCE0
CompositeCode
MicroPDF
UPCE0_2Supps
CouponCode
MicroQR
UPCE0_5Supps

DataMatrix
MSI
UPCE1
Discrete2of5
Mx25
UPCE1_2Supps
DOTCODE
NEC2OF5
UPCE1_5Supps
DutchPost
NetherlandKIX
UPUFICS
EAN128
OCR
US_Postal1
EAN13
PDF417
US_Postnet
EAN13_2Supps
Planet
USPlanet
EAN13_5Supps
Plessey
USPSS4CB

Не нашли что искали?

Задать вопрос в техническую поддержку

Тип «GO» в Mobile SMARTS

Последние изменения: 2024-03-26

GO является универсальным типом, который содержит в себе различные полезные методы. С помощью него можно выводить всплывающие сообщения, воспроизводить звуки, разбирать штрихкоды и др.

Методы

Метод
Описание
void Copy (data)
Копирует текст в буфер обмена* data — строка для копирования
string CreateUid ()
Генерирует уникальный строковый GUID
string DecryptForDevice (value)
Расшифровывает строку, зашифрованную на этом устройстве value — строка для дешифрования
string EncryptForDevice (value)
Зашифровывает строку. Расшифровать можно только на этом устройстве value — строка для шифрования
string DecryptForUser (value)
Расшифровывает строку, зашифрованную под текущим пользователем value — строка для дешифрования
string EncryptForUser (value)
Зашифровать строку с возможностью расшифровки только под текущим пользователем value — строка для шифрования
BarcodeData GetBarcodeData (barcode)
Возвращает объект для расширенной работы со штрихкодом barcode — строка, содержащая штрихкод
BarcodeData GetBarcodeData (barcode, type)
Возвращает объект для расширенной работы со штрихкодом barcode — строка, содержащая штрихкод type — тип штрихкода из списка
void HideWaitMessage ()
Закрыть сообщение об ожидании
string NormalizePhoneNumber (phoneNumber)
Преобразовывает номер телефона в вариант с +7 вместо 8, вырезает лишние символы phoneNumber — номер для нормализации
void OpenUrl (url)
Открыть URL в браузере

void Play (audio)
Воспроизводит звуковой файл audio — путь к файлу
void Play (audio, vibrate)
Воспроизводит звуковой файл audio — путь к файлу vibrate — настройка вибрации
void PlayError ()
Воспроизводит звук ошибки.
bool SendEmail (email, subject, text)
Отправляет письмо на указанную почту * email — адрес получателя subject — заголовок письма text — текст письма
bool SendEmail (email, subject, text, errors)
Отправляет письмо на указанную почту * email — адрес получателя subject — заголовок письма text — текст письма errors — прикрепить к письму логи с устройства
void SendErrorsLog ()
Отправка логов на сервер качества*
void Share (data)
Поделиться содержимым* data — путь к файлу или текст
void ShowBaloon (text, delay, down)
Вызывает всплывающее сообщение с указанным текстом text — текст сообщения delay — время отображения в мс down — выводить в нижней части окна *
void ShowErrorBaloon (text, delay, down)
Вызывает всплывающее сообщение об ошибке с указанным текстом. text — текст сообщения delay — время отображения в мс down — выводить в нижней части окна *
void ShowWaitMessage (text)
Показывает сообщение об ожидании поверх интерфейса text — отображаемое сообщение. Поддерживает верстку

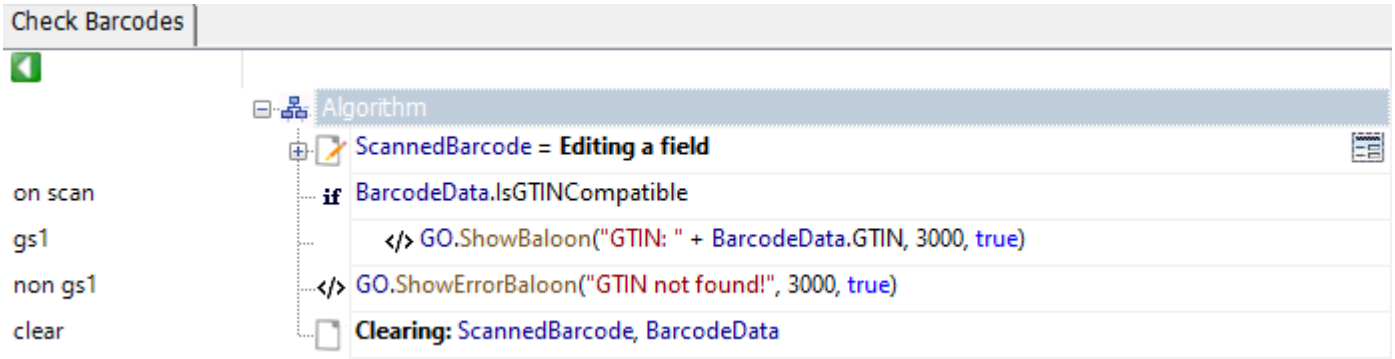
void ShowWaitMessage (text, canCancel)
Показывает сообщение об ожидании поверх интерфейса с настройкой возможности скрыть его по нажатию text — отображаемое сообщение. Поддерживает верстку canCancel — можно ли скрыть сообщение нажатием
void Sleep (ms)
Делает паузу в процессах на указанное кол-во мс ms — длительность паузы в миллисекундах

void Speak (text)
Произносит указанный текст* text — фраза для воспроизведения
void SpeakRubles (value)
Произносит указанную сумму с рублями и копейками* value — число для воспроизведения
void WriteError (text)
Заносит в лог сообщение с типом ошибка text — текст сообщения
void WriteInformation (text)
Заносит в лог сообщение с типом информация text — текст сообщения
void WriteWarning (text)
Заносит в лог сообщение с типом предупреждение text — текст сообщения
void WriteLog (text, logType)
Заносит в лог сообщение с указанным типом text — текст сообщения logType — тип записи в лог (info, warn, error, debug)

* работает только на Android-устройствах

Примеры использования

Задача 1: Добавить возможность по сканированию определить, содержит ли штрихкод GTIN.
Решение:

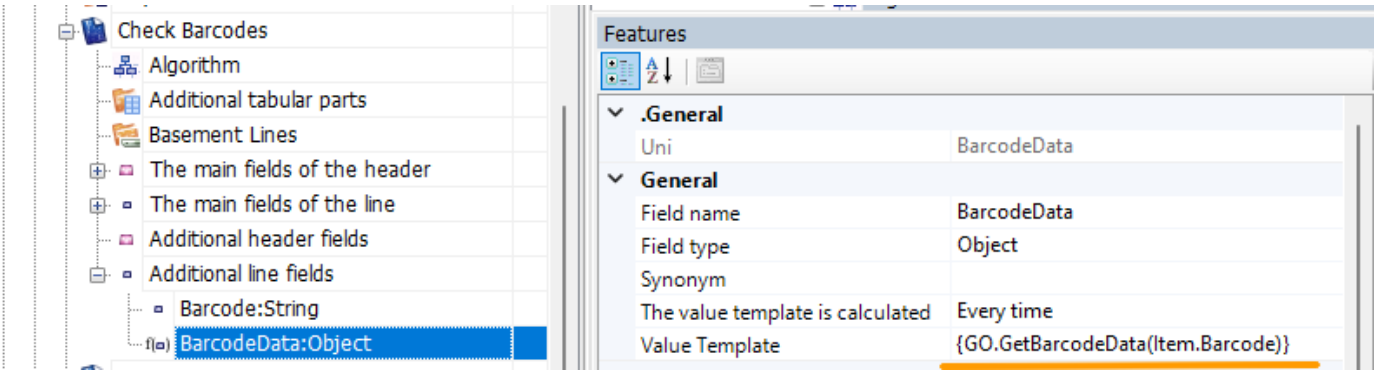


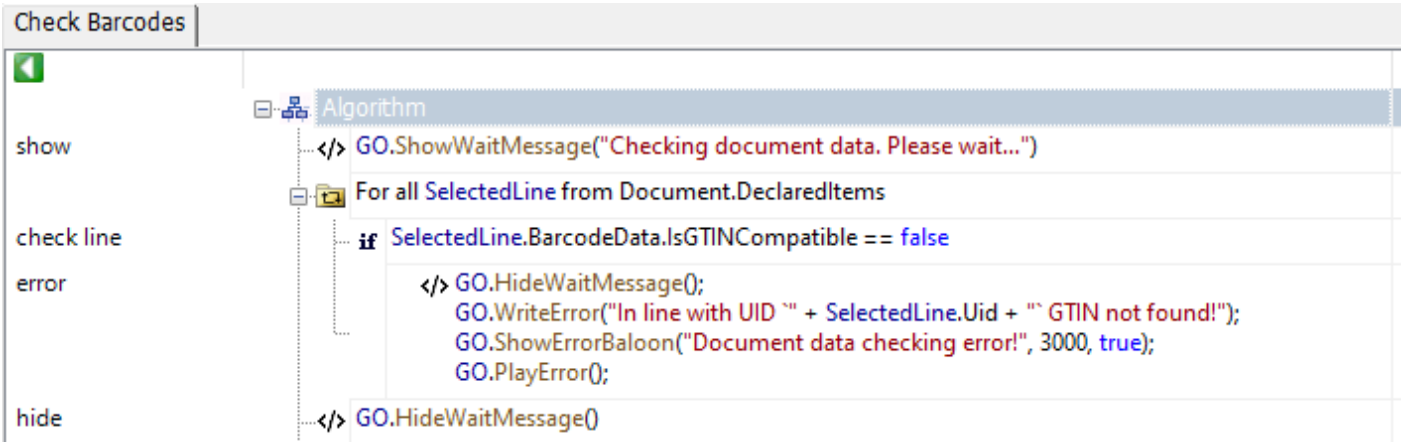
Разберем пример подробнее:

- В действии с именем метки «on scan» с помощью объекта типа BarcodeData проверяем, содержит ли отсканированный штрихкод в своем составе GTIN.
- В действии с именем метки «gs1» выводим всплывающее сообщение в нижней части экрана с информацией о GTIN'е
- В действии с именем метки «non gs1» выводим всплывающее сообщение об ошибке в нижней части экрана

Задача 2: В полях строк документа есть поле «Barcode», содержащее в себе штрихкод. Необходимо проверить, что в документе нет строк, в которых штрихкод не содержит GTIN.

Решение:





Разберем пример подробнее:

- **В структуру полей документа добавляем вычислимое поле с именем «BarcodeData».**
- **В шаблоне данного поля прописываем метод получения объекта типа BarcodeData.**
- **В действии с именем метки «show» выводим поп-ап, с уведомлением о том, что началась проверка строк документа. Так как строк может быть много проверка может занять некоторое время.**
- **Далее в цикле в действии с именем метки «check line» проверяем, содержит ли штрихкод в своем составе GTIN. Если содержит, переходим к следующему элементу.**
- **В действии с именем метки «error»**
- **Скрываем поп-ап**
- **Записываем ошибку в лог**
- **Выводим сообщение для пользователя ТСД.**
- **Воспроизводим звук ошибки**
- **Производим возврат документа**
- **Действие с именем «hide» выполняется если проверка документа выполнена без ошибок.**

Код примера

Список типов кодировок

AustraliaPost
EAN8
PosiCode

Aztec
GRIDMATRIX
Postals
BooklandEAN
GS1_128
Postnet
BPO
GS1_DATABAR
QR
CanadaPost
GS1_DATABAR_Expanded
RSS
ChinaPost
GS1_DATABAR_Limited
RSS14
Codabar
GS1_DATABAR_OMNIDIRECTIONAL
RSSExpanded
Codablock
GS1_Databar14
RSSLimited
Code11
GS1_DATAMATRIX
Straight2of5
Code128
GS1_QR
Straight2of5_IATA
Code128Emultion
HANXIN
Straight2of5_Industrial
Code16k
HK25
Telepen

Code32
IDTag
TLC39
Code32_PARAF
Interleaved2of5
TriopticCode39
Code39
ISBT128
UKPostal
Code49
JapanPost
UNKNOWN_TYPE
Code93
KoreanPost
UPCA
Composite_CCAB
Matrix2of5
UPCA_2Supps
Composite_CCC
MaxiCode
UPCA_5Supps
Composite_TLC39
MESA
UPCE0
CompositeCode
MicroPDF
UPCE0_2Supps
CouponCode
MicroQR
UPCE0_5Supps
DataMatrix
MSI
UPCE1
Discrete2of5
Mx25
UPCE1_2Supps

DOTCODE
NEC2OF5
UPCE1_5Supps
DutchPost
NetherlandKIX
UPUFICS

EAN128
OCR
US_Postal1
EAN13
PDF417
US_Postnet
EAN13_2Supps
Planet
USPlanet
EAN13_5Supps
Plessey
USPSS4CB

Не нашли что искали?

Задать вопрос в техническую поддержку

Тип «Server» в Mobile SMARTS

Последние изменения: 2024-03-26

Server — клиентский тип, с помощью которого осуществляется проверка доступности сервера MS, а также обмен служебными сообщениями. Данный тип недоступен в серверном контексте!

Свойства

Свойство	Описание
bool Available	Указывает, доступен ли сервер или работа идет в файловом режиме.
bool Online	Указывает, есть ли подключение к серверу в данный момент.
string Url	Возвращает url сервера, если он есть.

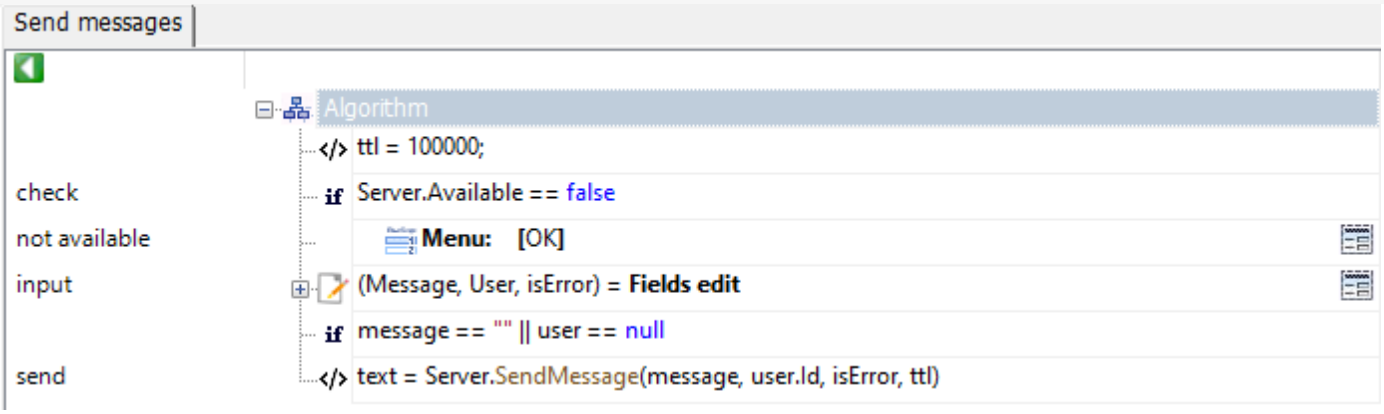
Методы

Метод	Описание
bool CheckConnection (timeout)	Проверить соединение с сервером. timeout — время ожидания ответа.
string SendMessage (text, to, isError, ttl)	Отправляет сообщение. text — текст сообщения to — ID пользователя или группы isError — отобразить как ошибку ttl — время жизни сообщения в секундах
string SendSystemMessage (text, isError, ttl)	Отправляет системное сообщение. text — текст сообщения isError — отобразить как ошибку ttl — время жизни сообщения в секундах

Пример использования

Задача: сделать так, чтобы любой пользователь ТСД мог отправить администратору уведомление о проблеме.

Решение:



Рассмотрим пример подробнее:

- В действии с именем метки «check» проверяем, что база развернута в серверном режиме.
- В действии с именем метки «input» вводим текст сообщения, выбираем пользователя, указываем, отобразить сообщение как ошибку или нет.
- В действии с именем метки «send» производится отправка сообщения на сервер, с которого сообщение будет доставлено пользователю.

Код примера

Не нашли что искали?

Задать вопрос в техническую поддержку

Тип «MathOperations» в Mobile SMARTS

Последние изменения: 2024-03-26

MathOperations – тип, содержащий метод, позволяющий округлять числа до нужного количества разрядов после запятой.

Доступные методы:

Имя метода	Описание
Round(val, digits, orient)	Округляет число до нужного количества разрядов после запятой

Примеры использования

Round

Есть 2 метода округления: до ближайшего чётного и дальше от нуля.

Например, в случае округления дальше от нуля, 3,75 округляется до 3,8, 3,85 округляется до 3,9, -3,75 округляется до -3,8 и -3,85 округляется до -3,9. Это ещё можно назвать округлением в большую сторону.

В случае же округления до ближайшего чётного, как 3,75, так и 3,85 округляются до 3,8 и -3,75 и -3,85 округляются до -3,8.

Если хотите использовать округление до ближайшего чётного, то в качестве переменной orient нужно передавать 0, для округления дальше от нуля - 1

Код:

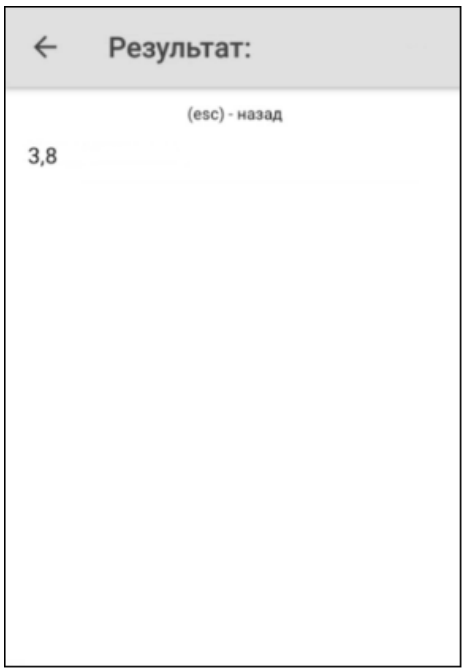
```
Число1 = MathOperations.Round(3.85,2,0);

Число2 = MathOperations.Round(3.85,2,1);
```

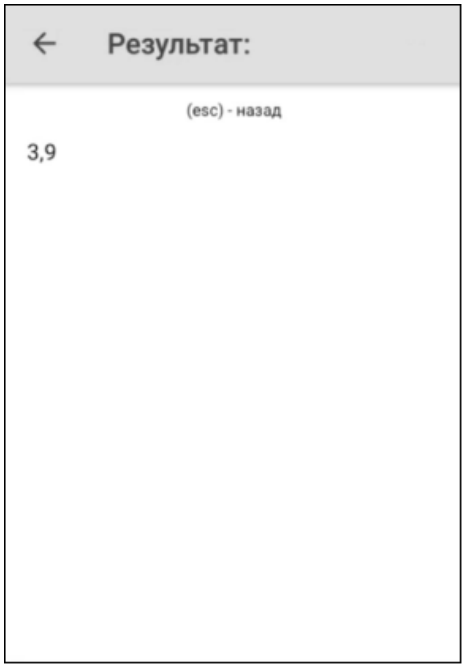
В первом случае округляем до ближайшего чётного, во втором дальше от нуля.

Результат:

До ближайшего чётного

Дальше от нуля





Не нашли что искали?

Задать вопрос в техническую поддержку

Тип «StringOperations» в Mobile SMARTS

Последние изменения: 2024-03-26

StringOperations – тип, содержащий методы, позволяющие работать со строками (обрезать, разбивать, форматировать и т.д.).

Доступные методы:

Имя метода	Описание
FillToWidth(s1, s2, totalWidth, c)	Объединяет две строки с разделителем между ними, дополняя итоговую строку до нужной длины
FromBase64(valueBase64)	Конвертирует строку из Base64 в обычную форму
Join(source, c)	Объединяет коллекцию строк, добавляя разделитель между ними
PadCenter(s, totalWidth, c)	Дополняет строку с обеих сторон до нужной длины, если она короче
PadLeft(s, totalWidth, c)	Дополняет строку слева до нужной длины, если она короче
PadRight(s, totalWidth, c)	Дополняет строку справа до нужной длины, если она короче
ParseByTemplate(value, template)	Позволяет распарсить строку по шаблону
Split(source, oneSymbolSplitter)	Разбивает строку на коллекцию подстрок согласно односимвольному разделителю. Если передан более длинный разделитель, то используется только первый символ
ToBase64(str)	Конвертирует строку в Base64 строку
WordWrap(text, width)	Форматирует строку переносами по словам под заданную длину строки

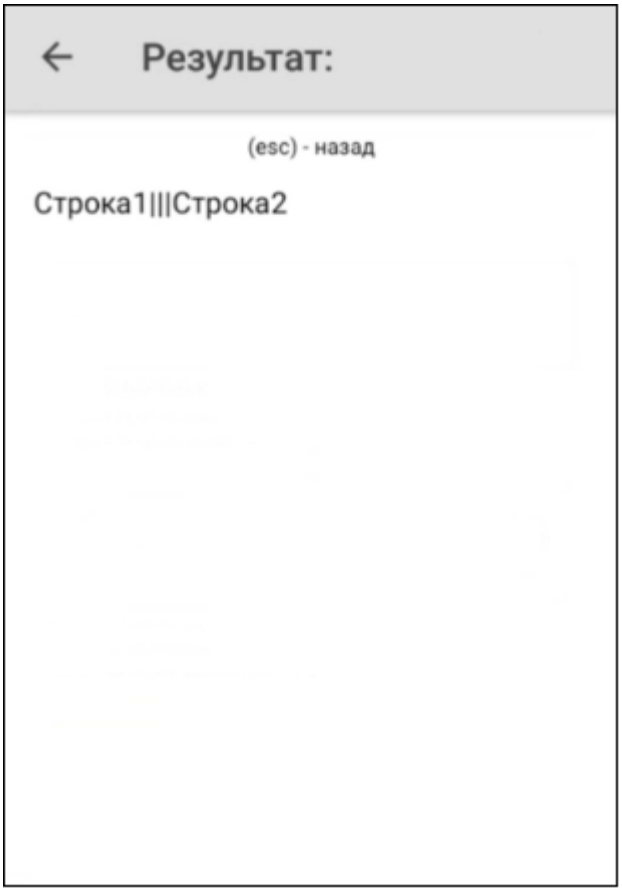
Примеры использования

FillToWidth

Код:

```
Строка = StringOperations.FillToWidth("Строка1","Строка2",17,"|");
```

Результат:



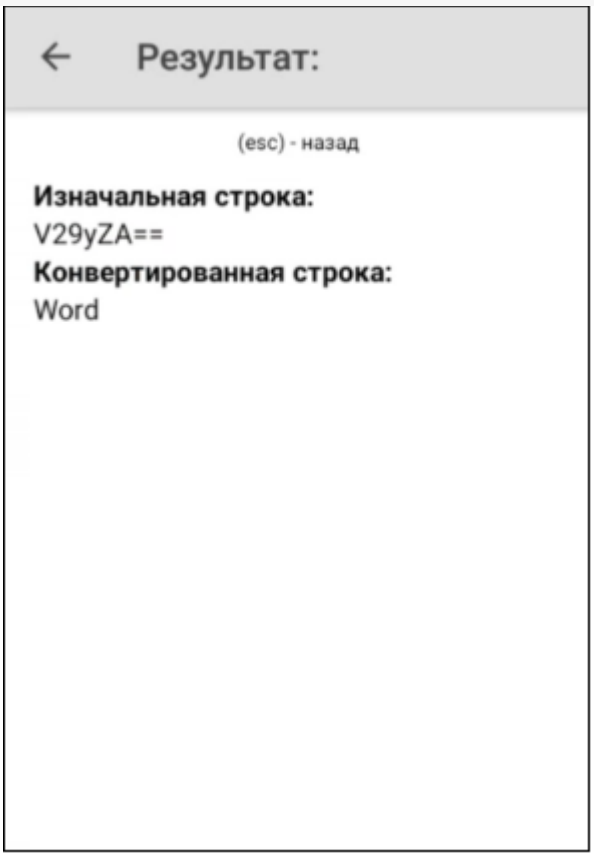
FromBase64

Код:

```
Строка = StringOperations.FromBase64("V29yZA==");
```

Строка «V29yZA==» расшифровывается как «Word»

Результат:



PadCenter

Код:

```
Строка = StringOperations.PadCenter("Текст",12,"|");
```

Очередность добавления символа: сначала справа, потом слева.

Результат:

PadLeft

Код:

```
Строка = StringOperations.PadLeft("Текст",10,"|");
```

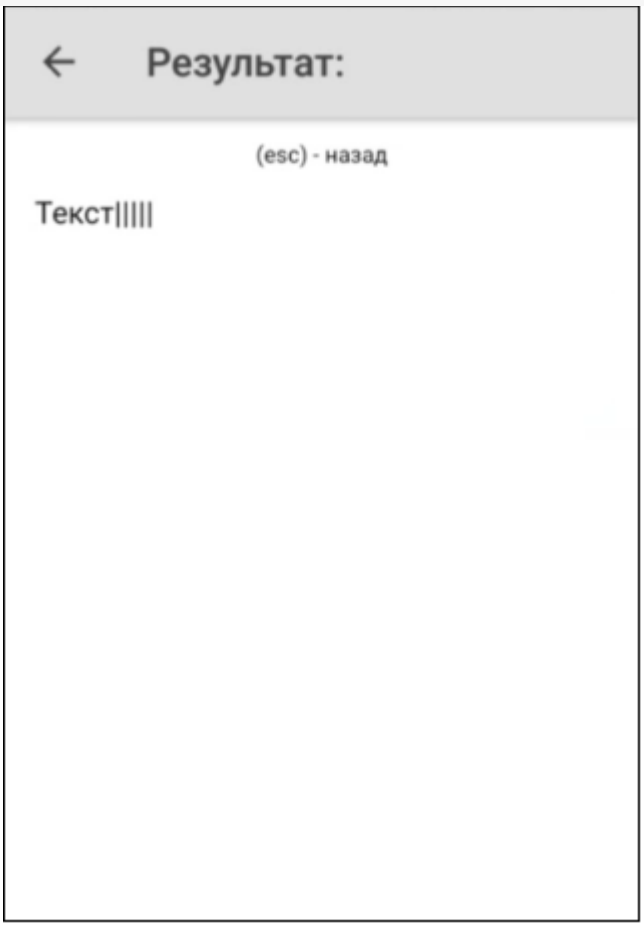
Результат:

PadRight

Код:

```
Строка = StringOperations.PadRight("Текст",10,"|");
```

Результат:



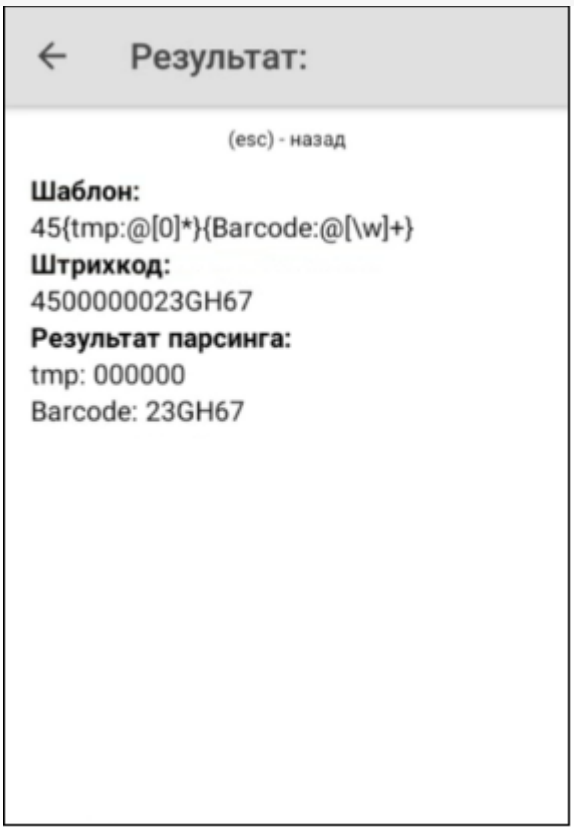
ParseByTemplate

Метод поддерживает любые шаблоны в конфигурации (regex, ячеек, номенклатуры и т.д.). Более подробно про регулярные выражения можно прочитать тут.

Код для регулярных выражений:

```
Строка = StringOperations.ParseByTemplate("4500000023GH67","45{tmp:@[0]*}
{Barcode:@[\\w]+}")
```

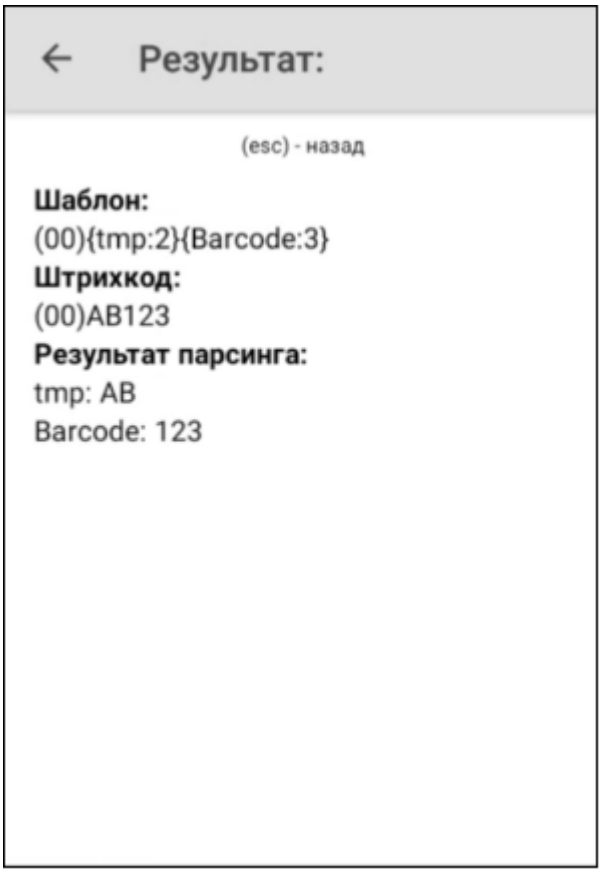
Результат:



Код для штрихкода:

```
Строка = StringOperations.ParseByTemplate("(00)AB123","(00){tmp:2}{Barcode:3}");
```

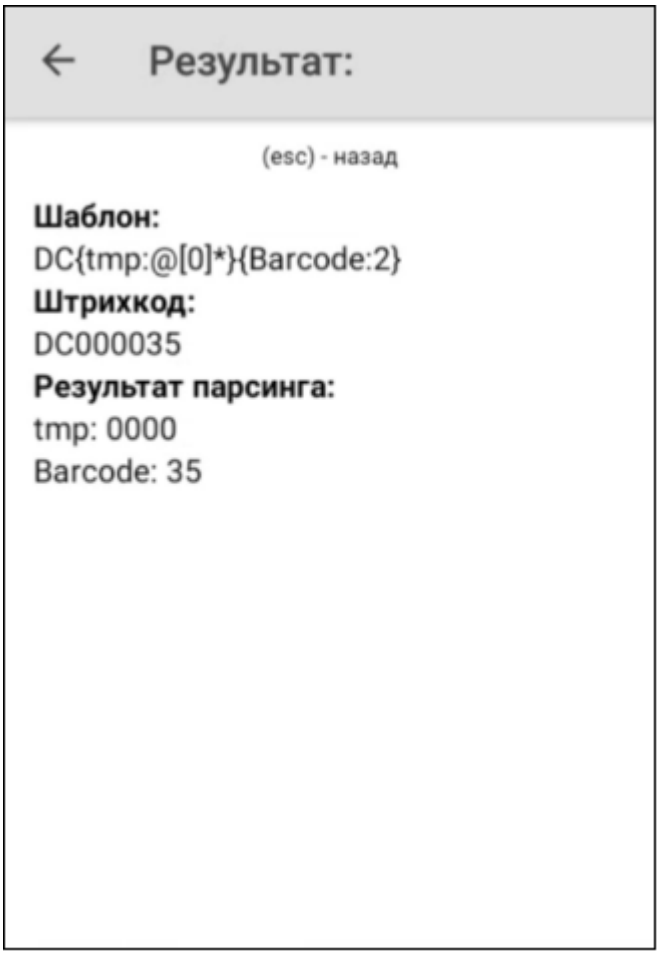
Результат:



Код для совмещённого случая:

```
Строка = StringOperations.ParseByTemplate("DC000035","DC{tmp:@[0]*}{Barcode:2}");
```

Результат:



Split

Код:

```
Строка = StringOperations.Split("Текст|для|разбиения","|");
```

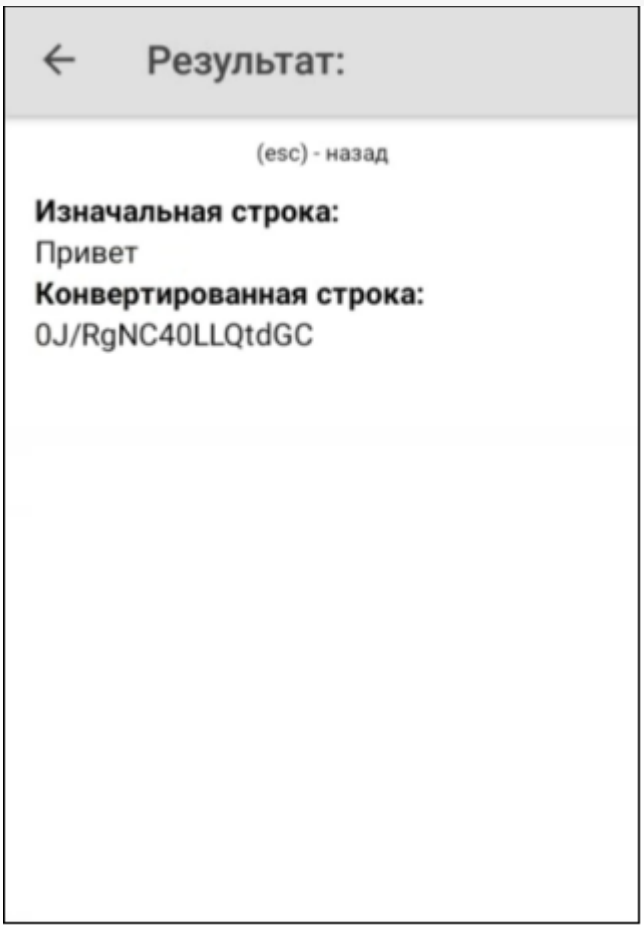
Результат:

ToBase64

Код:

```
Строка = StringOperations.ToBase64("Привет");
```

Результат:

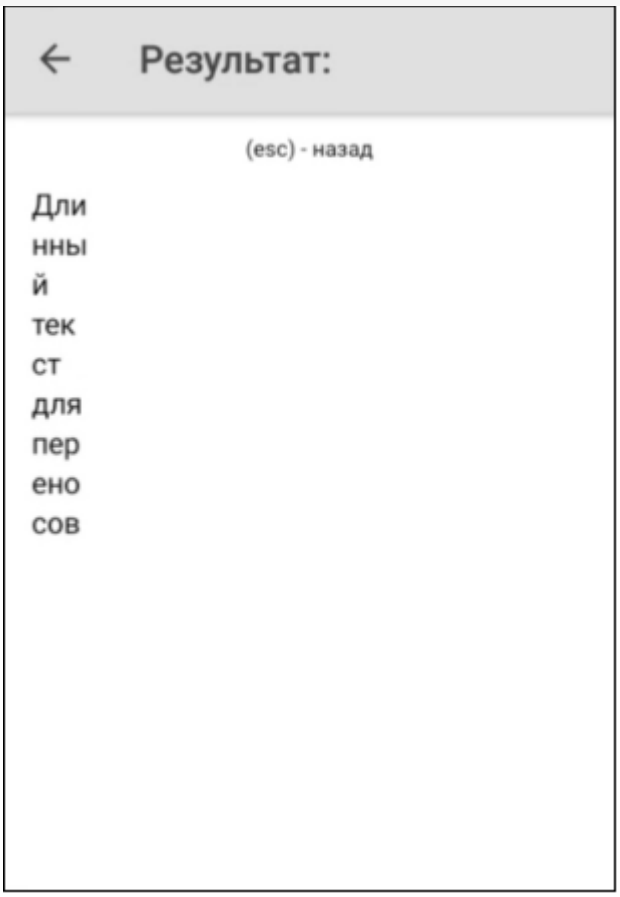


WordWrap

Код:

```
Строка = StringOperations.WordWrap("Длинный текст для переносов",3);
```

Результат:



Не нашли что искали?

Задать вопрос в техническую поддержку

Тип «UEObject» в Mobile SMARTS

Последние изменения: 2024-03-26

UEObject – тип, предназначенный для обработки данных, сохраненных в формате XML или JSON. Доступен в клиентском и серверном контексте выполнения. С помощью данного класса можно разбирать полученные данные, изменять структуру данных и собирать обратно в строку. Сериализация/ десериализация проводится без использования схемы/класса с метадатой.

Доступные поля:

Имя поля
Описание
Childs
Коллекция узлов и атрибутов объекта (UEObject)
HasValue
Есть ли значение
IsArray
Это массив
IsAttribute
Это атрибут
OName
Имя узла
OValue
Значение узла. Если строка загружена извне, тип по умолчанию — строка.
Type
Тип

Доступные методы:

Имя метода
Описание
UEObject()
Создаёт новый пустой UEObject (в случае XML создаёт атрибут <root>)
UEObject(string)
Создаёт новый UEObject с узлом внутри. Атрибут – название узла (для JSON нужно присвоить узлу значение)
FromXml(string)
Получить XML строку

FromJson(string, boolean)
Получить JSON строку. В качестве аргумента также передаётся наличие префикса «@» у атрибутов

GetAttribute(string)
Получить атрибут
GetNode(string)
Получить узел
GetNodes(string)
Получить несколько узлов
AddNode(string, object)
Добавить узел (строку). Аргументы – название узла, его значение (для JSON нужно добавлять значение, либо другой узел)
AddNode(UEObject)
Добавить узел (UEObject)
SetAttribute(string, object)
Добавить/заменить атрибут. Аргументы – название атрибута, его значение
Contains(string)
Содержит ли объект строку
AsBoolean()
Сохранить значение в булевой переменной
AsDateTime()
Сохранить значение в виде даты
AsDecimal()
Сохранить значение в виде десятичной дроби
AsDouble()
Сохранить значение в виде числа с плавающей запятой
AsInt32()
Сохранить значение в виде целого числа
AsString()
Сохранить значение в виде строки
ToXml(bool)
Сохранить объект в виде XML Аргумент – добавлять ли XML декларацию
ToJson(bool)
Сохранить объект в виде JSON Аргумент – добавлять ли префикс «@» к атрибутам

Примеры использования

Создание XML строки

К примеру, если мы хотим создать пустую XML строку, то это можно сделать так:

```
Объект = new Cleverence.Connectivity.UEObject();
   Строка = Объект.ToXml(true);
```

В этом коде мы создаём объект UEObject, конвертируем его в XML строку с помощью метода ToXml (с добавлением декларации), и заносим строку в переменную «Строка».

Если же нужно создать XML строку с начальным узлом, то нужно написать следующее:

```
Объект = new Cleverence.Connectivity.UEObject(«НовыйУзел»);
   Строка = Объект.ToXml(true);
```

В данном случае будет создан узел с именем «НовыйУзел» и пустым значением.

Результаты:

Без начального узла

С начальным узлом

Добавление узлов

К примеру, нам необходимо создать подобную XML строку:

```
<?xml version="1.0" encoding="UTF-8"?>
   <ObjectKey1><ObjectKey2>ObjectValue2</ObjectKey2></ObjectKey1>
```

Это можно сделать так:

```
Объект = new Cleverence.Connectivity.UEObject();
   Объект.AddNode("ObjectKey1", null).AddNode("ObjectKey2", "ObjectValue2");
   Строка = Объект.ToXml(true);
```

В объекте создаётся узел «ObjectKey1», который содержит в себе узел «ObjectKey2» со значением «ObjectValue2».

Результат:

Добавление атрибута

Атрибуты добавляются практически также, как узлы. Если в узел нужно добавить атрибут, то необходимо обратиться к узлу и написать метод «SetAttribute». К примеру:

```
Объект = new Cleverence.Connectivity.UEObject();
   Объект.AddNode("ObjectKey1", null).SetAttribute("ObjectAttribute", 15);
   Строка = Объект.ToXml(true);
```

В данном случае мы добавляем в узел «ObjectKey1» атрибут «ObjectAttribute» со значением 15:

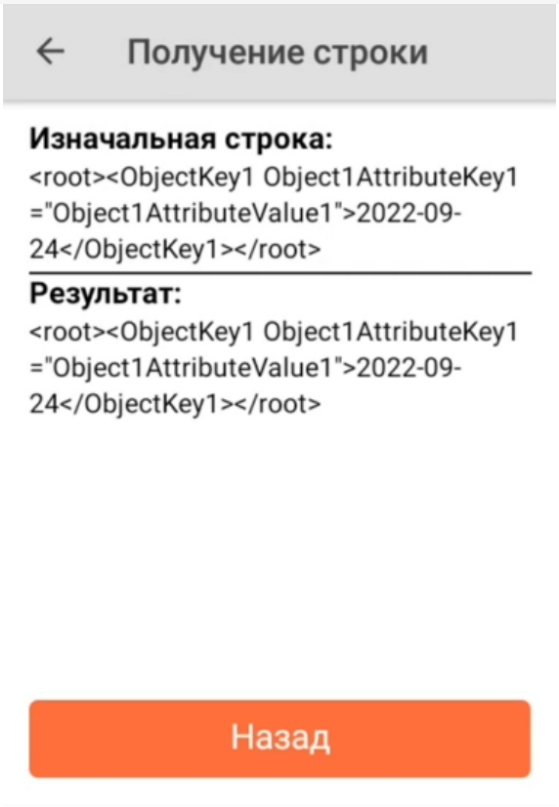
Получение строки XML

UEObject позволяет не только создавать свои строки, но и также получать их. К примеру, если у нас есть таблица, где хранятся XML строки, то можно вытащить оттуда строку и начать с ней работать. Сделать это можно таким образом:

```
Объект = new Cleverence.Connectivity.UEObject();
   Строка = select first (*) from XML where Item.Ид == "1";
   Объект.FromXml(Строка.Значение);
   Результат = Объект.ToXml(false);
```

Здесь мы вытаскиваем из таблицы XML строку с «Ид» равной 1 и записываем её в объект. В конце конвертируем объект в XML строку и записываем в переменную.

Результат:



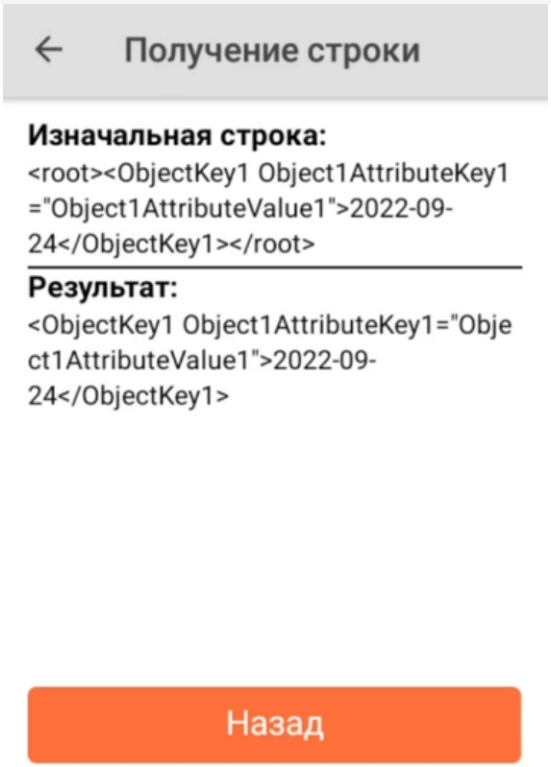
Результат работы кода приведён в нижней части.

Получение узла

Чтобы получить узел с его содержимым нужно написать следующий код:

```
Объект = new Cleverence.Connectivity.UEObject();
   Строка = select first (*) from XML where Item.Ид == "1";
   Объект.FromXml(Строка.Значение);
   Результат = Объект.GetNode("ObjectKey1").ToXml(false);
```

В метод GetNode нужно добавить название узла, который необходимо получить. Нам нужен узел «ObjectKey1». Если в узле, где ведётся поиск (в нашем случае «Объект») не будет найден узел с таким именем, то метод вернёт null.



В верхней части выводится вся строка

Получение полей узла

Если нужно получить какое-либо поле узла, то это можно сделать следующими способами:

```
Результат = Объект.GetNode("ObjectKey1").OValue;
```

```
Результат = Объект.ObjectKey1.AsDateTime();
```

```
Результат = Объект.GetNode("ObjectKey1").OName;
```

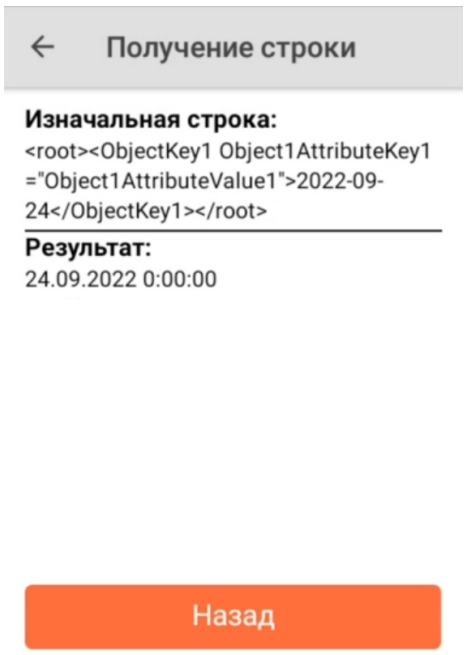
В первых двух случаях мы получаем значение узла. Отличие в том, что в первом случае мы получаем сырое значение узла (строку), а во втором в формате даты. И есть разница между тем, как идёт обращение к узлам: через метод GetNode или напрямую. В последнем же случае мы получаем имя узла.

Сырое значение

В формате даты

Имя узла


Получение строки
Изначальная строка:
<root><ObjectKey1 Object1AttributeKey1
="Object1AttributeValue1">2022-09-
24</ObjectKey1></root>
Результат:
2022-09-24
Назад


Получение строки
Изначальная строка:
<root><ObjectKey1 Object1AttributeKey1
="Object1AttributeValue1">2022-09-
24</ObjectKey1></root>
Результат:
24.09.2022 0:00:00
Назад


Получение строки
Изначальная строка:
<root><ObjectKey1 Object1AttributeKey1
="Object1AttributeValue1">2022-09-
24</ObjectKey1></root>
Результат:
ObjectKey1
Назад

Получение атрибута

Получить атрибут с его содержимым можно с помощью метода GetAttribute. Пример:

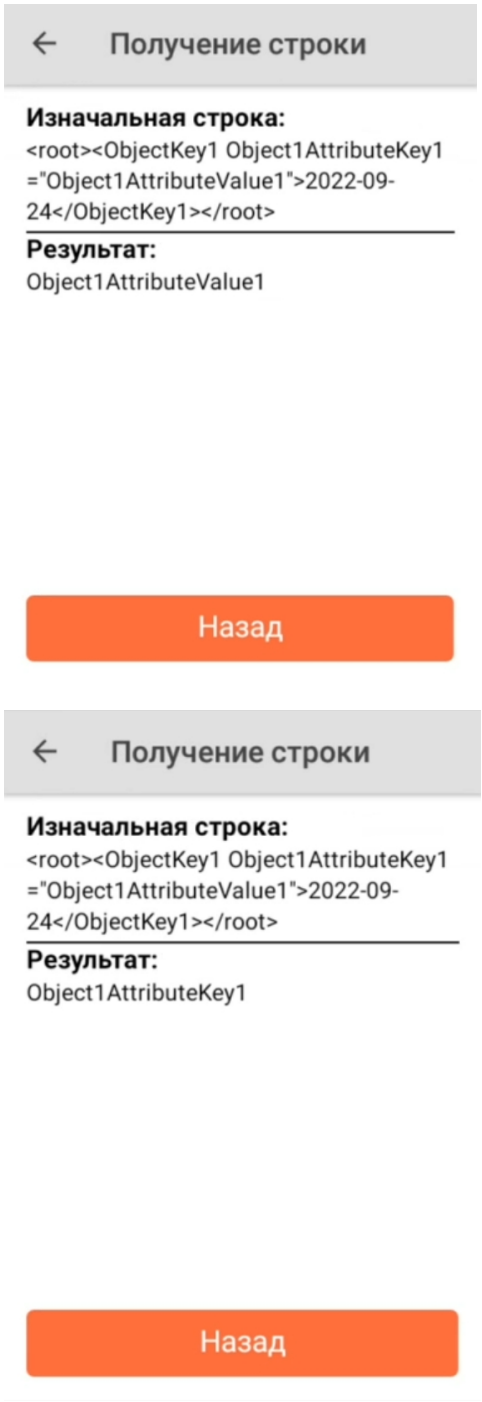
```
Результат = Объект.GetNode(«ObjectKey1».GetAttribute(«Object1AttributeKey1»).OValue;
```

```
Результат = Объект.GetNode(«ObjectKey1».GetAttribute(«Object1AttributeKey1»).OName;
```

В первом случае получаем значение атрибута, во втором его имя.

Значение

Имя


Получение строки
Изначальная строка:
<root><ObjectKey1 Object1AttributeKey1
="Object1AttributeValue1">2022-09-
24</ObjectKey1></root>
Результат:
Object1AttributeValue1
Назад

Получение строки
Изначальная строка:
<root><ObjectKey1 Object1AttributeKey1
="Object1AttributeValue1">2022-09-
24</ObjectKey1></root>
Результат:
Object1AttributeKey1
Назад

Использование select

UEObject позволяет использовать select. Возьмём в качестве примера предыдущий случай. Можно получить значение атрибута с помощью оператора select:

```
Значение = select (*) from Объект.ObjectKey1.Childs where Item.OName ==
"Object1AttributeKey1";
   Результат = Значение[0].OValue;
```

В данном случае мы обращаемся к коллекции Childs у узла ObjectKey1 и ищем в ней строку, у которой имя является именем атрибута «ObjectAttributeKey1». В полученной коллекции мы обращаемся к элементу с индексом 0 и получаем значение нашего атрибута.

Работа с JSON

Логика работы с JSON строками идентична с XML, но есть несколько важных моментов:

1. **При создании пустого объекта не добавляется узел «root»**
2. **Нельзя создать пустой объект UEObject с начальным узлом**
3. **В методах FromJson и ToJson есть булева переменная, отвечающая за наличие префикса «@» у атрибутов. Нужно для обозначения атрибутов в случае конвертации в/из XML.**
4. **Если у узла есть и значение и атрибут, то значение записывается в «#text»**

Примеры:

```
Объект = new Cleverence.Connectivity.UEObject();
   Объект.AddNode("ObjectKey1", null).SetAttribute("ObjectAttribute", 15);
```

Создаём объект с узлом «ObjectKey1» и атрибутом «ObjectAttribute» со значением 15.

ToJson(true)

ToJson(false)


```
Объект = new UEObject();
   Объект.AddNode("ObjectKey1", "ObjectValue1").SetAttribute("ObjectAttribute", "Attribute");
   Строка = Объект.ToJson(true);
```

Создаём объект с узлом «ObjectKey1» у которого есть и значение и атрибут. Добавляем префикс для атрибута.

Взаимные конвертации

XML и JSON строки можно конвертировать между собой. Самое главное – при конвертации из JSON в XML задавать атрибутам префиксы.

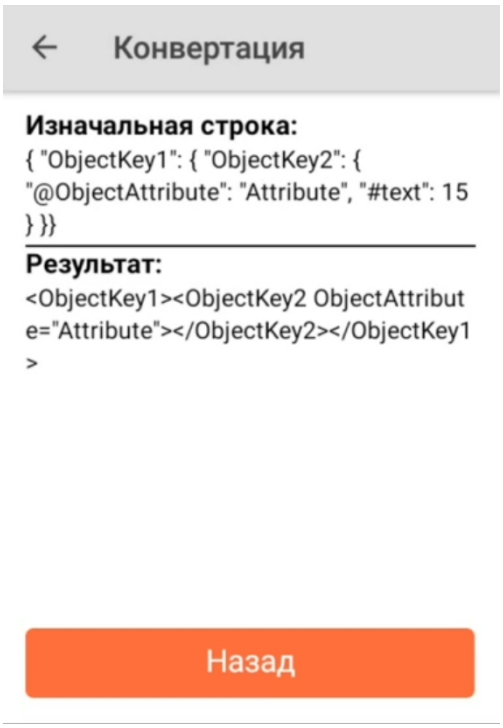
```
Объект = new UEObject();
   Объект.AddNode("ObjectKey1", "ObjectValue1").AddNode("ObjectKey2",
15).SetAttribute("ObjectAttribute", "Attribute");
   Строка = Объект.ToXml(false);
   Объект2 = new Cleverence.Connectivity.UEObject();
   Объект2.FromXml(Строка);
   Строка2 = Объект2.ToJson(true);
```

Здесь мы создаём объект с 2-я узлами и атрибутом, в переменную «Строка» записываем XML строку, создаём новый пустой объект и помещаем туда нашу XML строку. В конце, конвертируем 2-ой объект в JSON строку и записываем в «Строка2».

Результаты:

Аналогично поступаем в случае конвертации из JSON в XML (не забывая обозначать атрибуты префиксами).

```
Объект = new Cleverence.Connectivity.UEObject();
   Объект.AddNode("ObjectKey1", "ObjectValue1").AddNode("ObjectKey2",
15).SetAttribute("ObjectAttribute", "Attribute");
   Строка = Объект.ToJson(true);
   Объект2 = new Cleverence.Connectivity.UEObject();
   Объект2.FromJson(Строка, true);
   Строка2 = Объект2.ToXml(false);
```



Не нашли что искали?

Задать вопрос в техническую поддержку

Транзакции в таблицах в Mobile SMARTS

Последние изменения: 2024-03-26

Транзакция представляет собой набор действий, необходимых для изменения состояния таблицы. Важная особенность транзакции — изменения по транзакции могут быть либо полностью применены, либо полностью отменены. Как пример — перевод денежных средств с одного счета на другой. Не должно возникнуть ситуации, когда деньги списались с одного счета, но не зачислились на другой.

На текущий момент транзакции применимы только к глобальным таблицам. За изменения состояния документа отвечает стек действий.

Так как таблицы — отдельные сущности, для каждой таблицы необходимо открывать свою транзакцию. Методы работы транзакций должны быть вызваны у типа таблицы, в которую вносятся изменения.

Методы

Метод	Описание
bool BeginTransaction ()	Инициализирует транзакцию на основании текущего состояния таблицы. Данные, заносимые в таблицу, доступны только внутри данной транзакции.
bool CommitTransaction ()	Применяет изменения, вносимые данной транзакцией.
bool RollbackTransaction ()	Отменяет изменения данной транзакции.

Особенности работы

При исполнении серверного кода транзакция может быть открыта неявно. Такая транзакция открывается в момент занесения данных в таблицу. Закрытие транзакции происходит в момент завершения серверной операции. Это сделано для того, чтобы в случае ошибки исполнения операции все занесенные данные были сброшены к состоянию таблицы на момент внесения изменений. При этом, возможность явного открытия транзакции так же присутствует.

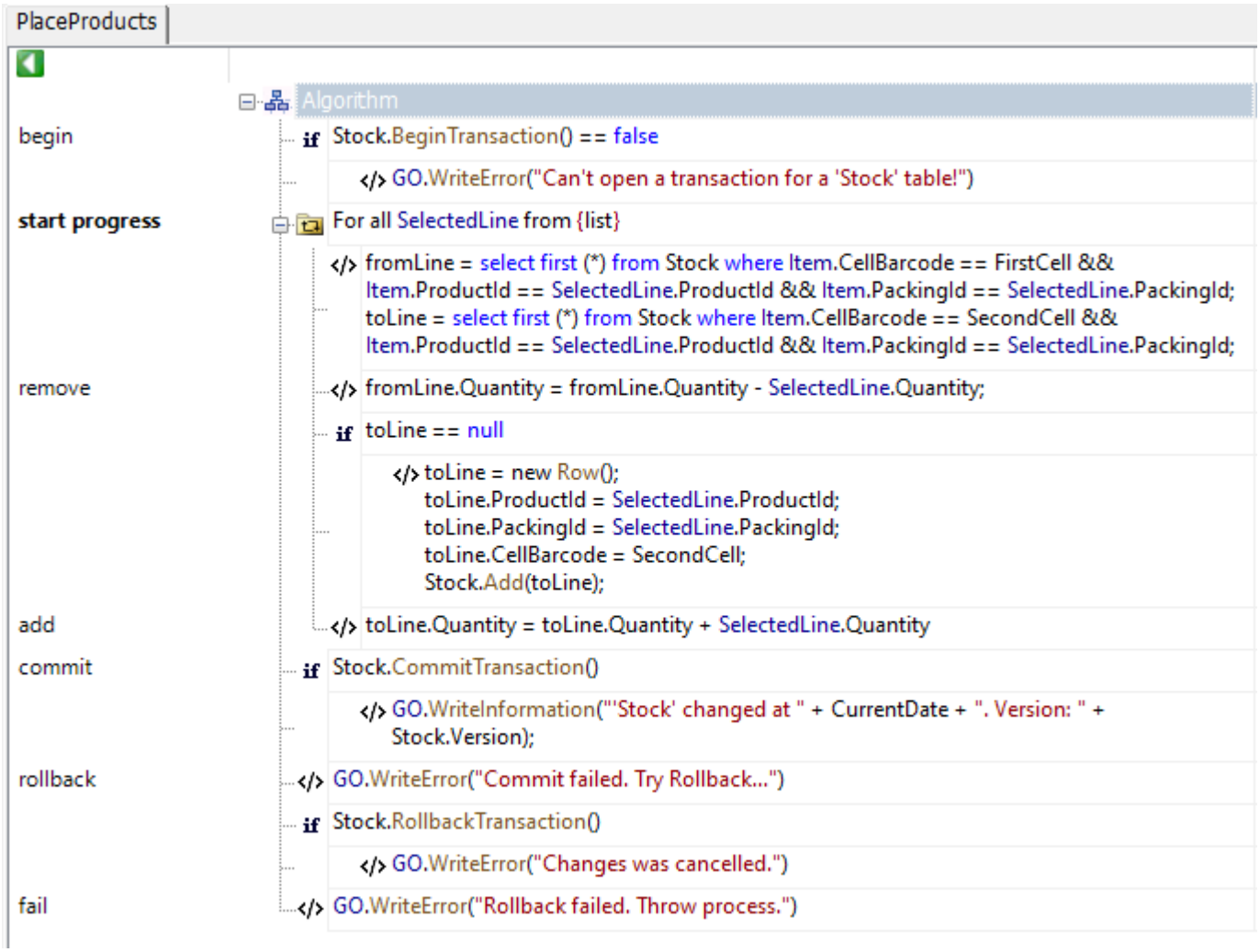
При исполнении клиентского кода необходимо явно открывать транзакцию. При этом, транзакция автоматически закрывается при достижении визуального действия либо действия записи в документ.

Пример использования

Задача: Существует глобальная таблица Stock содержащая в себе информацию о количестве товаров в наличии и их размещении. Необходимо переместить 5 единиц товара из одной ячейки в другую в реальном времени.

Решение:

* структура таблицы Stock



Рассмотрим решение подробнее:

- В действии с именем метки «begin» вызываем метод открытия транзакции для таблицы Stock
- В действии с именем метки «start progress» обрабатываем коллекцию строк объектов типа PackedProduct, а именно:
  - В действии с именем метки «remove» вычитаем количество в строке товара в исходной ячейке
  - В действии с именем метки «add» добавляем вычтенное ранее количество

- После обработки всех строк вызываем метод применения транзакции к таблице в действии с именем метки «commit»
- В случае успеха заносим запись в лог о том, что транзакция успешно проведена и завершаем выполнение операции.
- В случае ошибки применения транзакции вызываем метод отката изменений транзакции в действии с именем метки «rollback», после чего прерываем выполнение операции.

Код примера

Не нашли что искали?

Задать вопрос в техническую поддержку