# Программирование пользовательского интерфейса приложения Mobile SMARTS

Последние изменения: 2024-03-26

## Возможности по конфигурированию

Каждое визуальное действие, с которым связана экранная формочка на ТСД, имеет свои индивидуальные настройки, с которыми приходится ознакомиться для того, что составить представление о возможностях по конфигурированию той или иной формы.

Но что общее для всех форм - так это замечательная возможность применения шаблонов текстов и математических выражений. Шаблоны текстов - это возможность указать практически в любом месте вместо статического текста кусок псевдо-HTML, который вычисляется динамически и позволяет оформить/раскрасить по своему усмотрению любой элемент формы. Возможность указать в шаблоне элементы изображений позволяет максимально просто задавать иконки, картинки товаров, фотографии сотрудников и т.п. вещи.

## Набор экранных форм

На текущий момент список возможных форм ТСД в Mobile SMARTS следующий:

- **Жесткие формы, составная часть основного приложения**

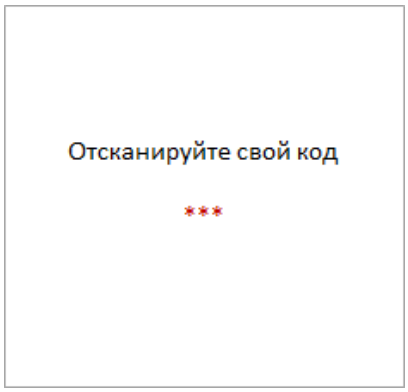
  Для примера - показаны четыре основные формы. Помимо них есть форма ввода пароля администратора, форма блокировки экрана и т.п.

  Часть форм опциональные, т.е. в зависимости от настроек приложения они могут отсутствовать.

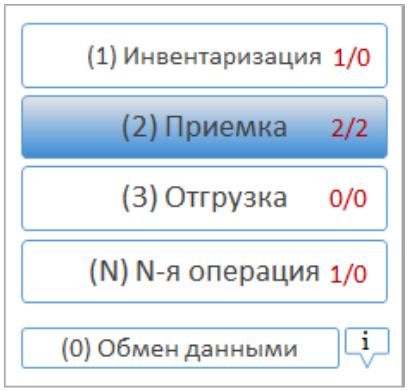
  Несмотря на то, что формы называются «жесткими», в последних версиях андроид-клиента Mobile SMARTS присутствует возможность регулировать цвет, форму кнопок, содержимое «шторки» и еще некоторые параметры. Подробнее об этом читайте в статье «Доработка визуального оформления главного меню «Магазина 15» для ОС Android».

Вход в программу

Главное меню




Опциональное окно. Не отображается, если пользователь в системе только 1 и у него нет пароля.




Обязательное окно. Но набор операций настраиваем для каждой группы пользователей. Можно настроить порядок и вид кнопок.

Выбор документа для работы

Создание нового документа

Опциональное окно, т.к. документы можно создавать автоматически при выборе операции в главном меню. Также можно разрешать/запрещать ручной выбор, выбор сканированием и т.п.

Опциональное окно. Можно запретить создавать документы на ТСД. Можно настроить автосоздание документа при выборе операции в главном меню.

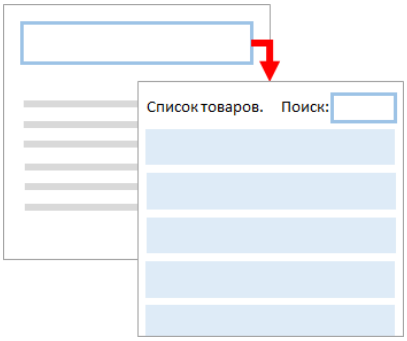
- **Формы действий, из которых можно строить алгоритмы обработки**

**Порядок, количество, частично внешний вид и особенно поведение - настраиваемы в среде разработки Mobile SMARTS.**

Выбор номенклатуры

Поле ввода со списком

Поля ввода




**Действие «Выбор номенклатуры»** любым из следующих способов (настраиваемо): сканированием, вводом цифр вручную, выбором вручную из списка, поиском по значению переменной, поиском по части названия.

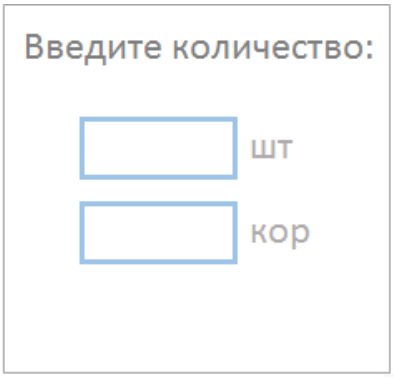
**Действие «Редактирование поля»**, с возможностью отображения списка вариантов для выбора значения.

**Действие «Редактирование полей»**, с возможностью управлять отображением поля и форматом вводимых значений.

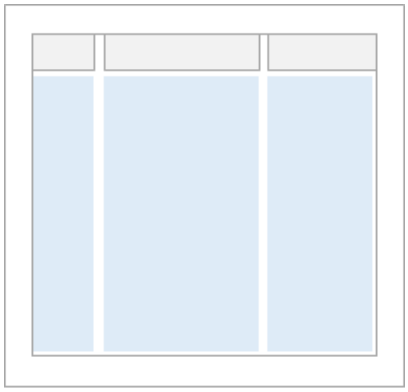
**Ввод количества**

**Список или таблица**

**Сообщение**




**Действие «Ввод количества»** номенклатуры. Позволяет редактировать сразу в нескольких единицах измерения (в зависимости от номенклатуры), есть встроенный калькулятор.



**Действие «Просмотр записей»**, с возможностью задать источник данных, колонки, формат выводимых значений, реакцию на выбор строки.

**Действие «Сообщение»**, выводит информацию в духе MessageBox (не путать с сообщениями, посылаемыми на ТСД в духе ICQ).

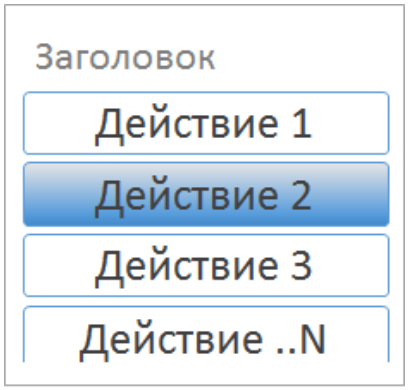
### Ошибка

### Выбор «Да/Нет»

### Меню

**Действие «Сообщение»**, в варианте «Отображать как ошибку (красным) = Да».

**Действие «Выбор Да/ Нет»**, позволяет задать пользователю вопрос.




Действие меню из кнопок, подходит как для собственно меню, так и для вопросов с несколькими вариантами ответа.

### Произвольный отчет

### Фотографирование

Действие произвольного окна с таблицами, текстами, кнопками. Возможен просмотр с прокруткой.

Действие фотографирования. Возможность сделать фотографию, если поддержка такой функции реализована для конкретного устройства.

интерфейс

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Стиль клиентского приложения Mobile SMARTS на ТСД

Последние изменения: 2024-03-26

Данная статья применима только к ТСД на ОС Windows CE!

**Mobile SMARTS поддерживает работу со многими ТСД, у которых разные размеры экранов, на которых кнопки и текст выглядят по-разному (в зависимости от разрешения экрана). Для простоты настройки отображения текстов и кнопок для конкретного ТСД предусмотрен стиль клиентского приложения (начиная с версии 2.6.7.13).**

**Изменение стиля выполняется в файле MobileSMARTS.exe.config, который расположен в папке приложения на ТСД (например, для терминалов Motorola, \Application\MobileSmarts\MobileSMARTS.exe.config).**

**Стиль задается в конфигурационной группе <style> </style>.**

```
...
<add key = "screenMode" value = "Color"/>
...
<style>
<add key = "menuButtonHeight" value = "34" />
<add key = "LargeFont" value = "Tahoma;12pt;Bold" />
<add key = "NormalFont" value = "Tahoma;11pt;" />
<add key = "ListFont" value = " Tahoma;10pt;" />
<add key = "ButtonForeColor" value = "Black" />
<add key = "ButtonFocusForeColor" value = "Black" />
<add key = "ListFocusForeColor" value = "White" />
<add key = "ListFocusBackColor" value = "LightSteelBlue" />
</style>
...
```

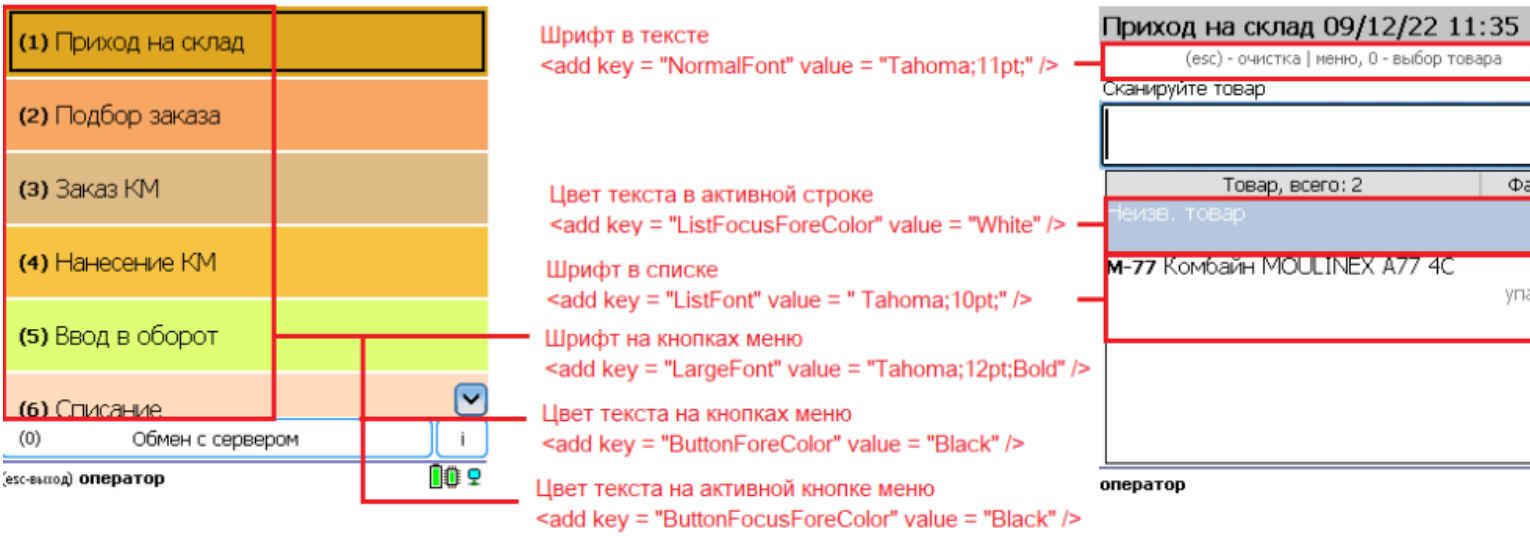
## Возможные настройки

| Стиль | Краткое описание | Значение по ум |
|---|---|---|
| screenMode | | |
| Цветное или черно-белое отображение кнопок. Подробнее >> | | |
| Color | | |

| |
|---|
| menuButtonHeight |
| Минимальная высота кнопки в пикселах. Подробнее >> |
| 34 |
| LargeFont |
| Шрифт в кнопках меню. Подробнее >> |
| Tahoma;12pt;Bold |
| NormalFont |
| Шрифт в тексте. Подробнее >> |
| Tahoma;11pt; |
| ListFont |
| Шрифт в списках. Подробнее >> |
| Tahoma;10pt; |
| ButtonForeColor |
| Цвет текста в кнопках меню. Подробнее >> |
| Black |
| ButtonFocusForeColor |
| Цвет текста активной (выделенной) кнопки меню. Подробнее >> |
| Black |
| ListFocusForeColor |
| Цвет текста активной (выделенной) строке в списке. Подробнее >> |
| White |

| |
|---|
| ListFocusBackColor |
| Цвет фона активной (выделенной) строки в списке. Подробнее >> |
| LightSteelBlue |

Цвет текста может быть задан по разному, можно указать название цвета на английском языке (например, red, blue, green) value = «Black», или задать код цвета в шестнадцатеричном виде value = «#0000FF». Таблицу цветов можно посмотреть здесь.
Шрифт задается в виде строки Имя;Размер;Стиль.
Имя: обычно используется Arial или Tahoma
Размер: число с добавлением единицы измерения pt. Обычные значения 12pt10pt и т. д.
Стиль: Regular — обычный, Bold — жирный текст, Italic — курсив
Примеры:
Tahoma;12pt
Arial;10pt;Bold
Tahoma;10pt;Italic




## Отображение кнопок

### Минимальная высота кнопки

<addkey = «menuButtonHeight» value = «34» /> - 34 пиксела минимальная высота кнопки, установленная по умолчанию.

В случае если надпись на кнопке не помещается, кнопка автоматически подстроится под размер (станет выше, а текст перенесется на новую строку).

## Шрифт и размер текста на кнопках меню

Строка <addkey = «LargeFont» value = «Tahoma;12pt;Bold» /> задает шрифт в кнопках меню на терминале по умолчанию.

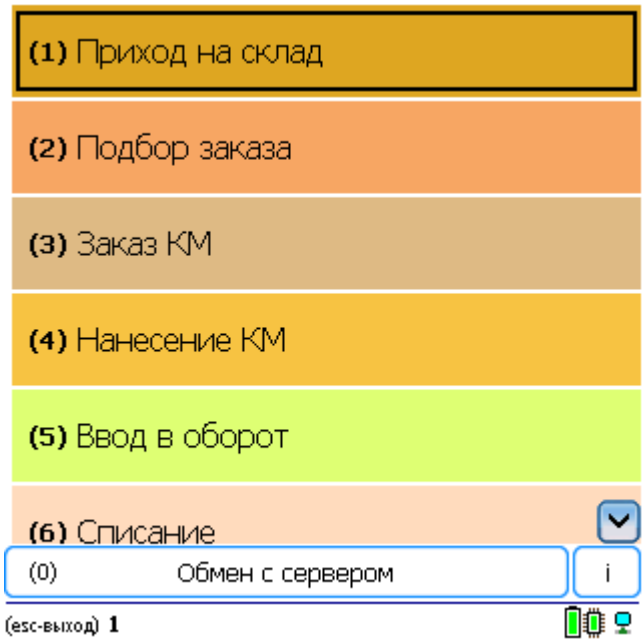
Tahoma; - означает назначенный по умолчанию шрифт;

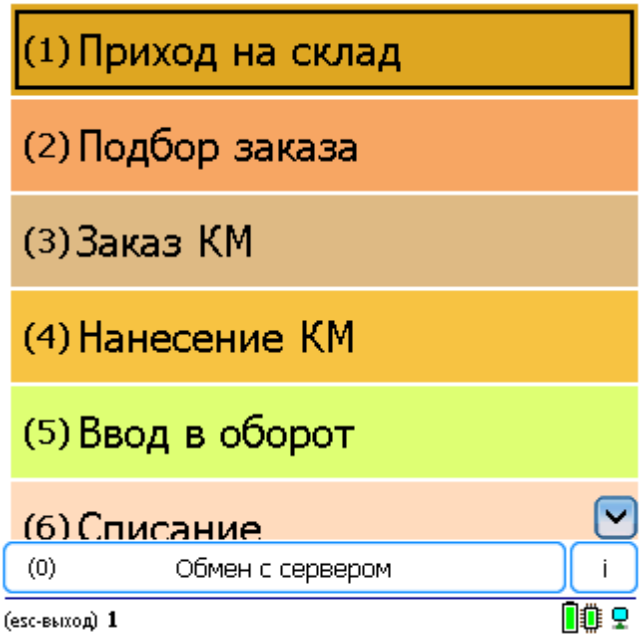
12pt; - размер шрифта;

Bold — выделение жирным.

<add key = «LargeFont» value = «Tahoma;12pt;Regular» />

<add key = «LargeFont» value = «Cadara;16pt;Bold» />



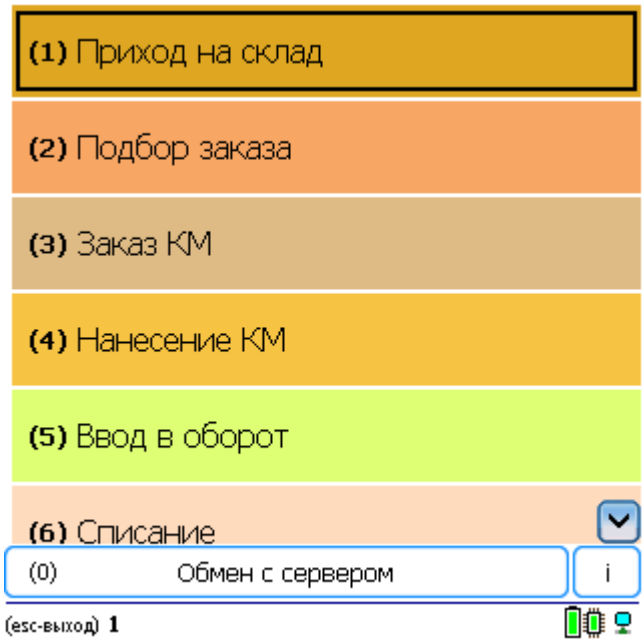



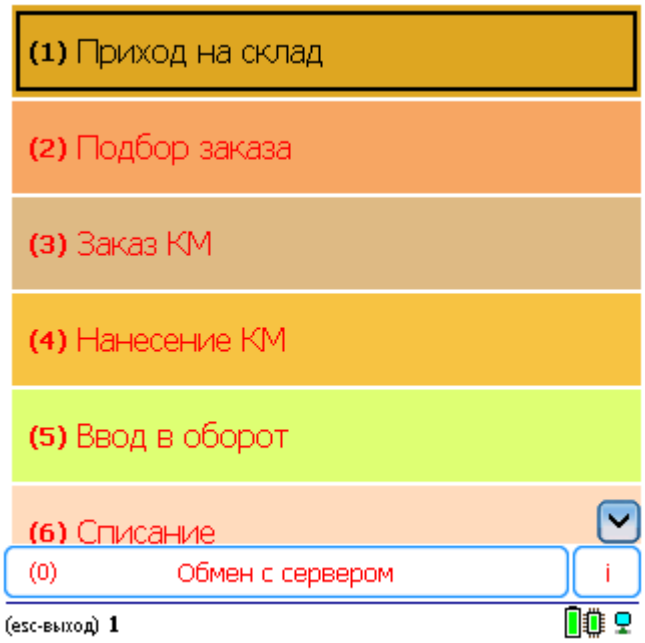

## Цвет текста на кнопках меню

По умолчанию цвет текста в кнопках меню задается в строке <addkey = «ButtonForeColor» value = «Black» />. Можно указать название цвета на английском языке (например, red, blue, green) color="blue», или задать код цвета в шестнадцатеричном виде color="#0000FF» (таблицу кодов цвета смотрите здесь).

<add key = «ButtonForeColor» value = «Black» />

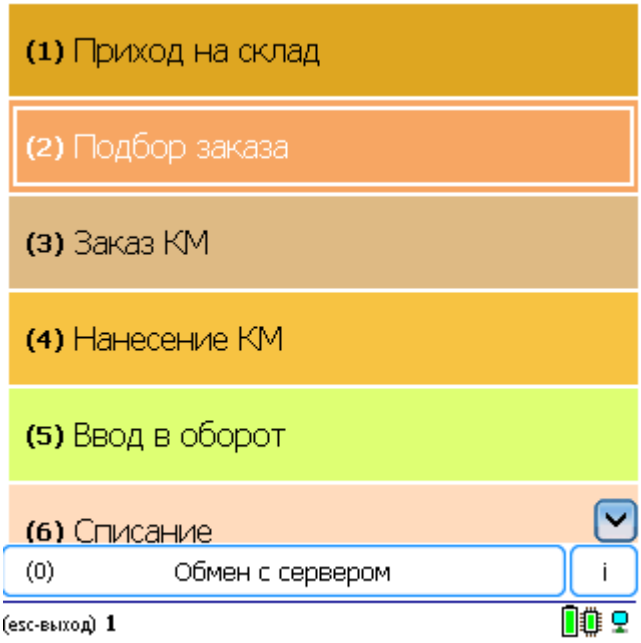
<add key = «ButtonForeColor» value = «Red» />




## Цвет текста активной кнопки меню

Для изменения цвета активной (выделенной) кнопки меню, установленный по умолчанию, в строке <addkey = «ButtonFocusForeColor» value = «White» /> задаем нужный нам цвет (например, White). Теперь во всех меню на ТСД текст в активной кнопке будет белым.




## Цветное или черно-белое отображение кнопок

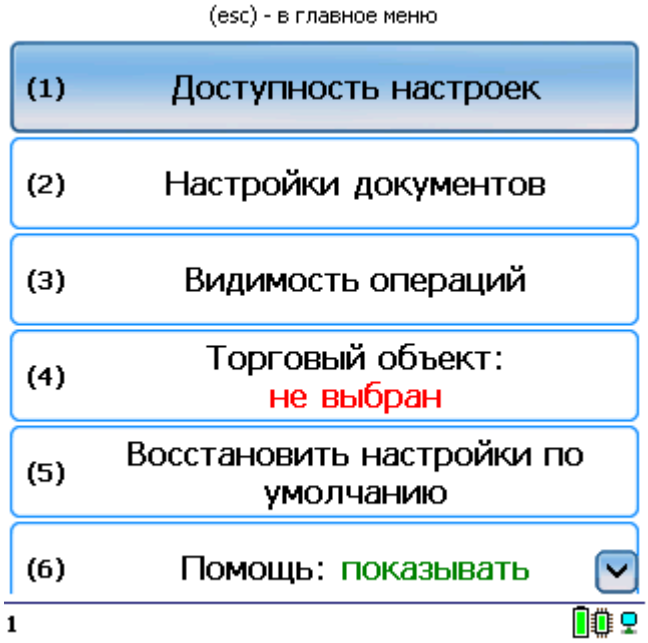
По умолчанию кнопки отображаются цветные. Для более корректного отображения кнопок на ТСД с черно-белым экраном в строке < add key = «screenMode» value = «Color"/> можно изменить «Color» на «BW».

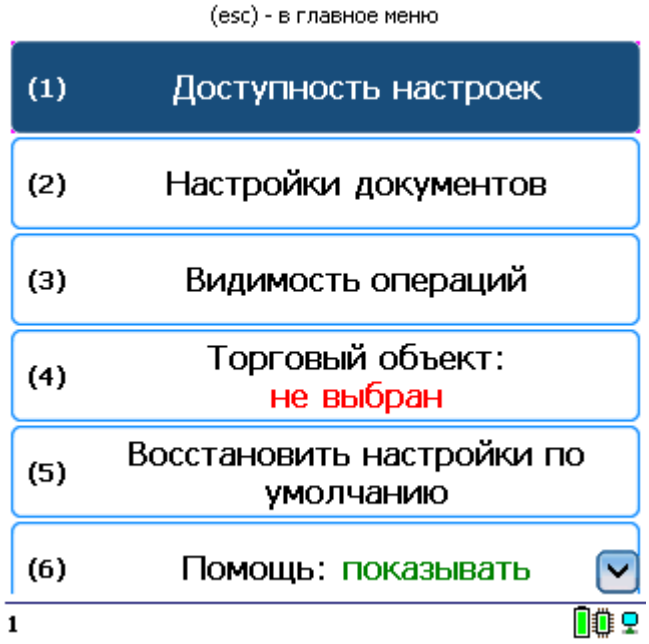
<add key = «screenMode» value = «Color"/>

Цветное отображение кнопок

<add key = «screenMode» value = «BW"/>

Черно-белое отображение кнопок

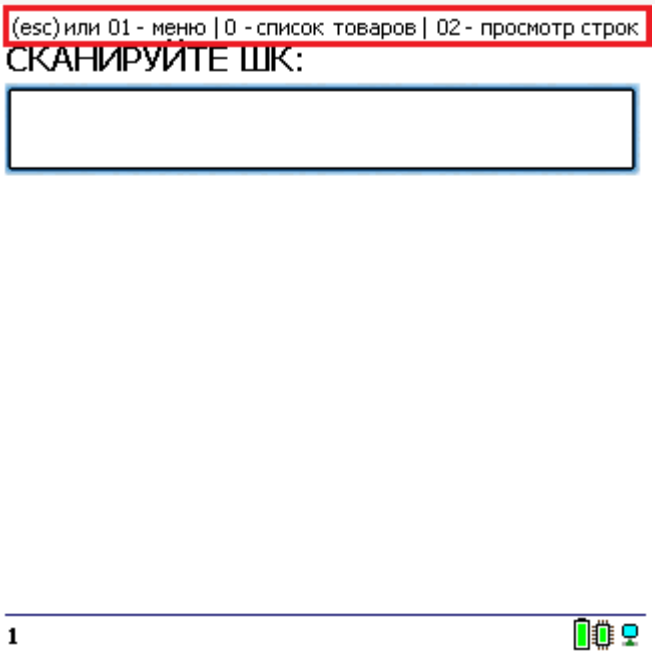



## Отображение текста

### Шрифт и размер в тексте

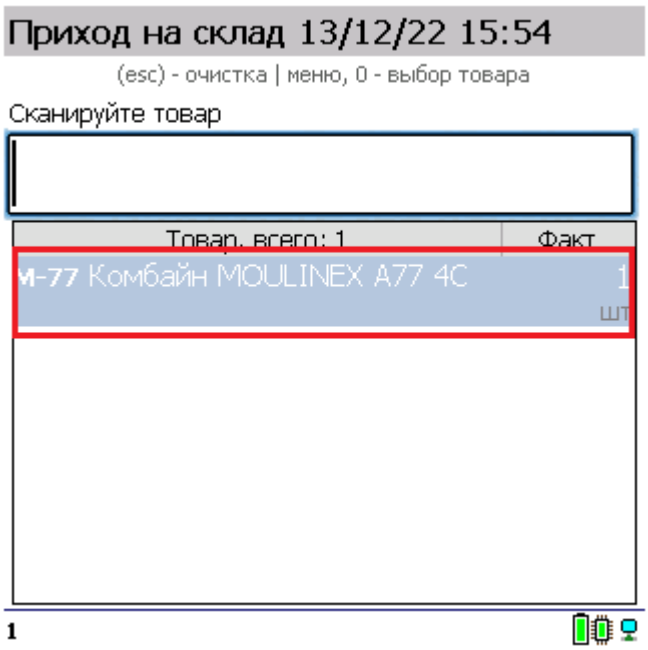
Строка <addkey = «NormalFont» value = «Tahoma;11pt;» /> - задает шрифт и его размер в тексте на терминале (например, в шапке). По умолчанию задается шрифт Tahoma и размер size="11».




## Шрифт и размер в списках

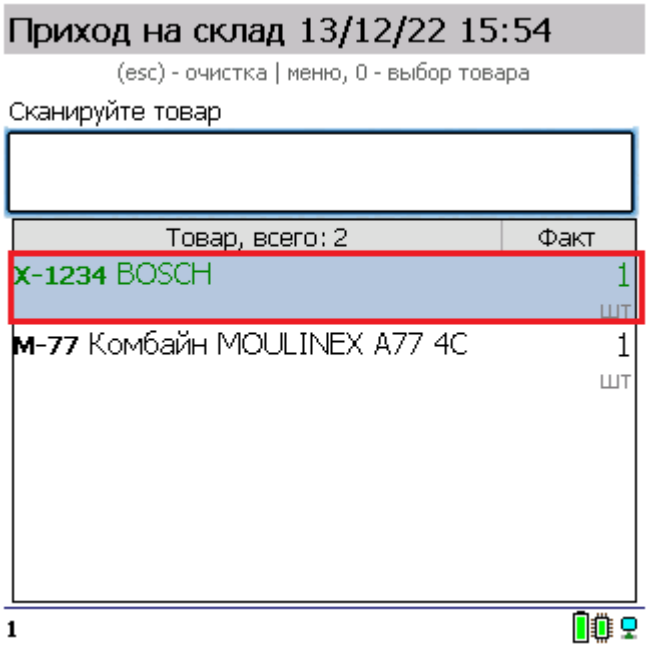
Строка <addkey = «ListFont» value = «Tahoma;10pt;» /> - задает шрифт и его размер в списках.

Если не задан (<add key = «ListFont» value = «» />), то берется шрифт и размер из строки «NormalFont».



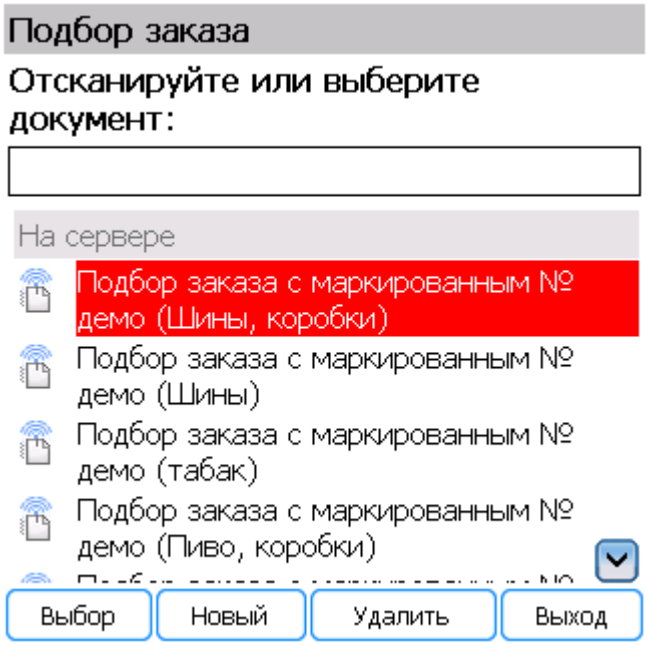

## Цвет текста в активной строке

Кроме размера шрифта в списке по умолчанию в строке <add key = «ListFocusForeColor» value = «#008000» /> задается цвет текста в активной (выделенной) строке (например, #008000 — зеленый). Теперь во всех списках на ТСД текст в активной строке будет зеленым (кода цветов смотрите здесь).




## Цвет фона в активной строке

**Кроме цвета текста активной (выделенной) строки можно изменять цвет фона назначенного по умолчанию в строке <add key = «ListFocusBackColor» value = «#FF0000» />. Теперь во всех списках на ТСД фон в активной строке будет красным (кода цветов смотрите здесь).**




клиентское приложение, интерфейс, Win CE

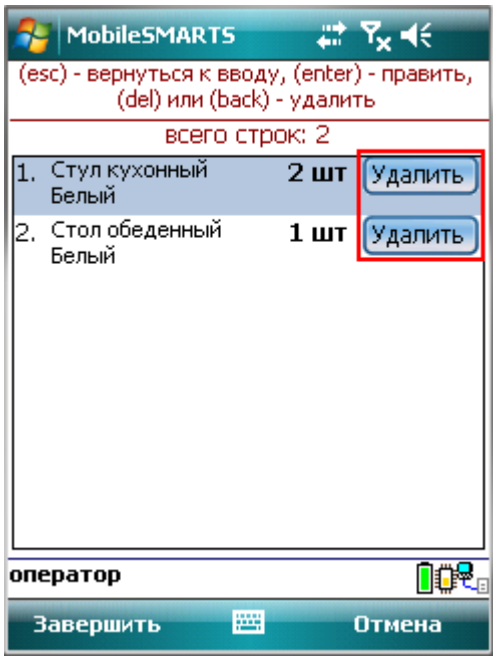
## Не нашли что искали?

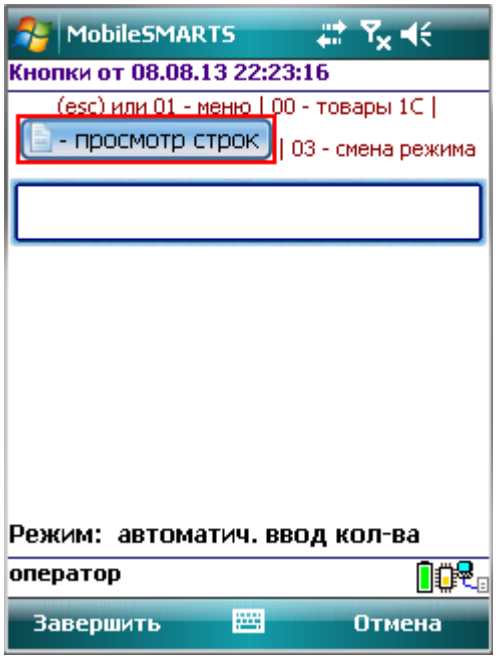
Задать вопрос в техническую поддержку

# Добавление дополнительных управляющих кнопок в приложении Mobile SMARTS

Последние изменения: 2024-03-26

Начиная с версии 2.6.7.11, в Mobile SMARTS предусмотрена возможность добавлять дополнительные управляющие кнопки в визуальных действиях. При нажатии на кнопку будет происходить переход на другое действие, которое было задано в её описании, аналогично переходу по быстрым клавишам или управляющим штрихкодам, добавленных с помощью обработчиков визуальных действий. Кнопки можно вставлять практически в любом месте интерфейса, где может быть настроен вывод текста: в шапках действий, на кнопках меню, строках «подвала», в отображении списков строк документа и т. д.




Выражение, которое задает кнопку, имеет следующий вид:

[HTML]

```
<button direction="..." width="..." height="..." align="..." enabled="{...}">Текст кнопки или картинка</button>
```

**Для перехода по нажатию кнопки на какое-нибудь действие необходимо указать атрибут direction="имя действия для перехода».**

[HTML]

```
<button direction="имя действия для перехода">Текст кнопки или картинка</button>
например,
<button direction="Сканирование">Текст кнопки или картинка</button>
```

**Для задания размеров кнопки в пикселях или процентах используются атрибуты width=»...» (ширина) и height=»...» (высота). Размеры кнопки задавать не обязательно. Когда размеры не указываются, тогда они задаются автоматически по ширине и высоте текста или изображения.**

[HTML]

```
<button width="..." height="...">Текст кнопки или картинка</button>,
например,
<button width="220" height="25">Текст кнопки или картинка</button>
<button width="80%" height="10%">Текст кнопки или картинка</button>
<button width="80%" height="30">Текст кнопки или картинка</button>
```

**Текст относительно кнопки можно выравнивать по центру (center), верхней (top) или нижней (bottom) границе. По умолчанию выравнивается по нижней границе (если атрибут align не указан).**

[HTML]

```
<button align="...">Текст кнопки или картинка</button>,
например,
<button align="top">Текст кнопки или картинка</button>
<button align="center">Текст кнопки или картинка</button>
<button align="bottom">Текст кнопки или картинка</button>
```

**В Mobile SMARTS предусмотрена возможность при определенных условиях делать кнопки активными (можно нажать кнопку) или пассивными (на кнопку нажать нельзя).**

[HTML]

<button enabled="{условие}">Текст кнопки или картинка</button>,
например,
<button enabled="{Document.CurrentItems.Count>0}">Текст кнопки или картинка</button>
если введенное условие (Document.CurrentItems.Count>0) выполняется, то кнопка будет активна, если не выполняется, то на кнопку нажать нельзя.

**Также есть возможность полностью скрыть кнопку при соблюдении введенных условий.**

[HTML]

<button visible="{условие}">Текст кнопки или картинка</button>,
например,
<button visible="{Document.CurrentItems.Count>0}">Текст кнопки или картинка</button> если введенное условие (Document.CurrentItems.Count>0) выполняется, то кнопка отображается, если нет, то не отображается.

**Разрешается использовать форматирование, как текстов кнопки, так и изображений.**

интерфейс, действия, управляющие кнопки

## Не нашли что искали?

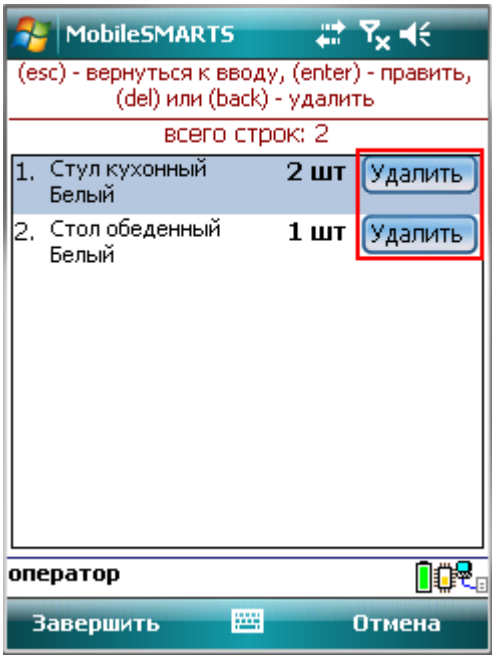
Задать вопрос в техническую поддержку

# Примеры создания дополнительных управляющих кнопок в приложении Mobile SMARTS

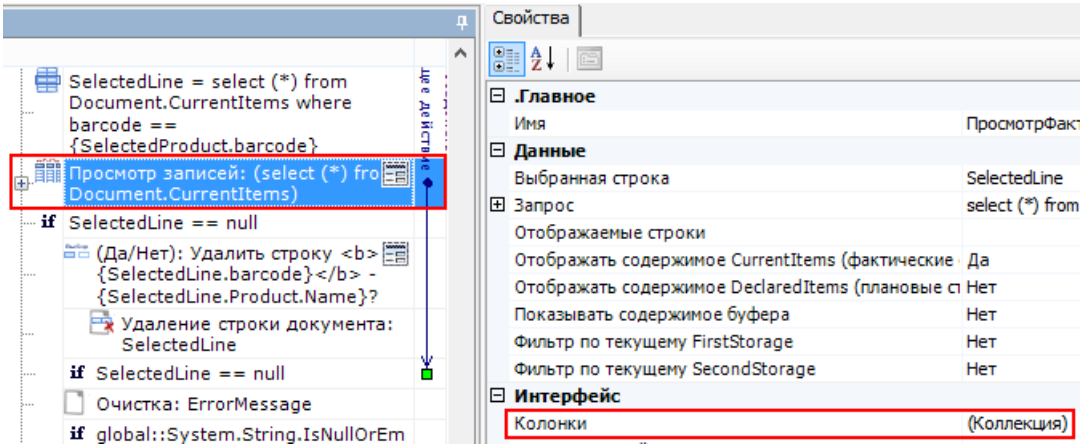
Последние изменения: 2024-03-26

## Пример 1 | Кнопка задана в строке текстом

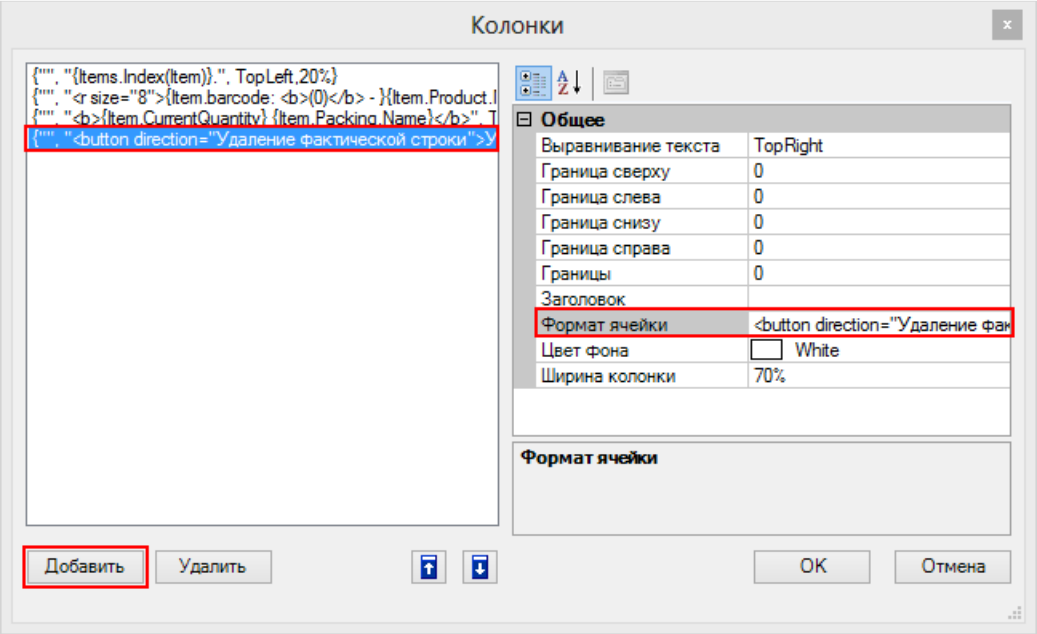
Посмотрим, как сделать кнопку в строке при просмотре записей.




В действие «Просмотр записей» добавим новую колонку.



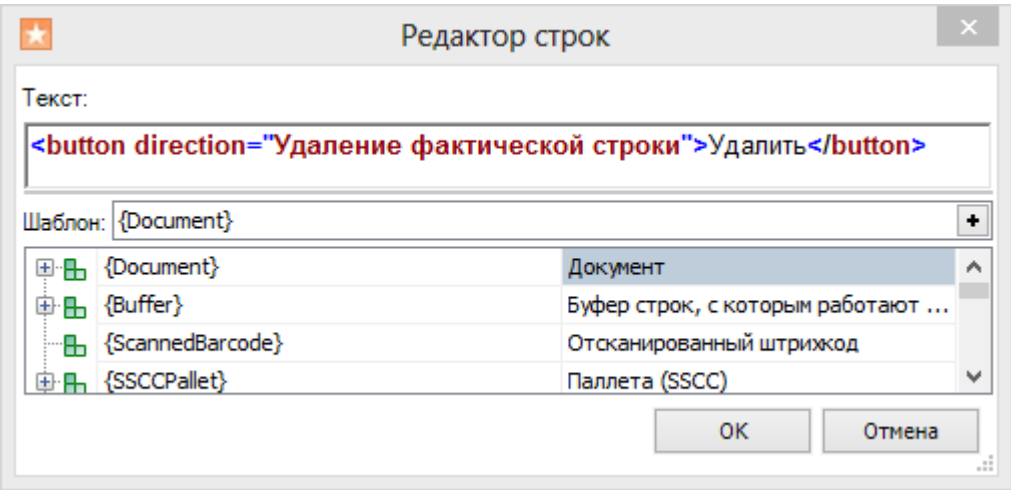

В формат ячейки напишем выражение для добавления кнопки.



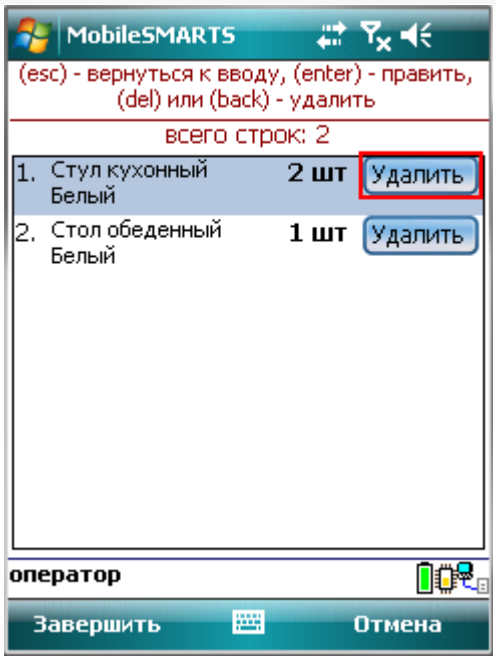

**Наше выражение для добавления кнопки выглядит так:**

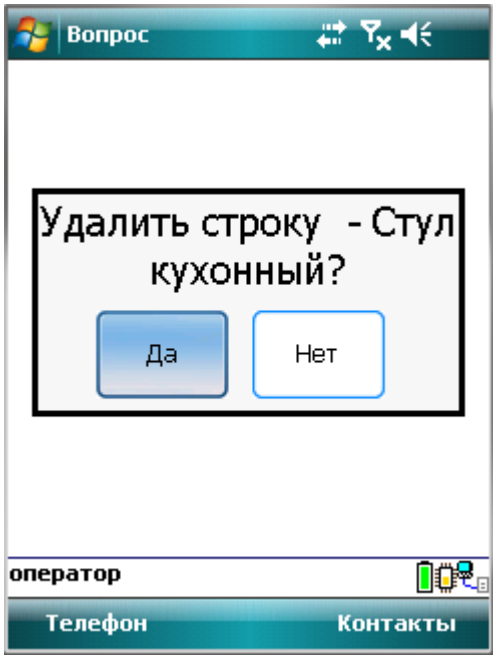
[HTML]

<button direction="Удаление фактической строки">Удалить</button>,
где
Удаление фактической строки – имя действия, на который будет выполнен переход по нажатию кнопки;
Удалить – надпись на кнопке.



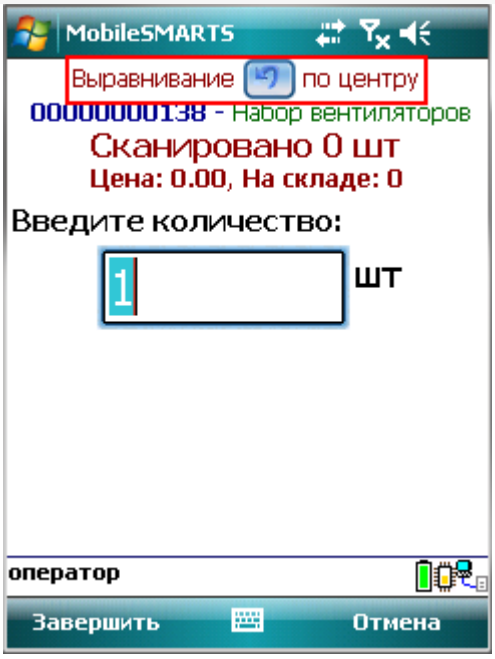

**Теперь при нажатии на кнопку будет выполняться переход на действие по удалению строки.**



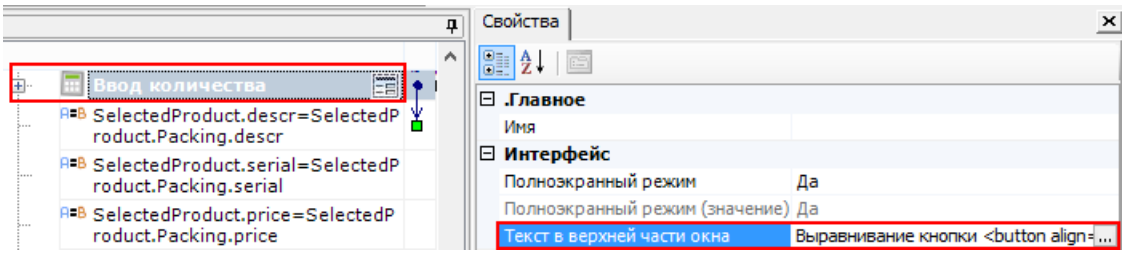




## Пример 2 | Кнопка задана изображением

Рассмотрим, как сделать кнопку в верхней части окна сканирования с изображением и выравниванием по центру относительно текста.



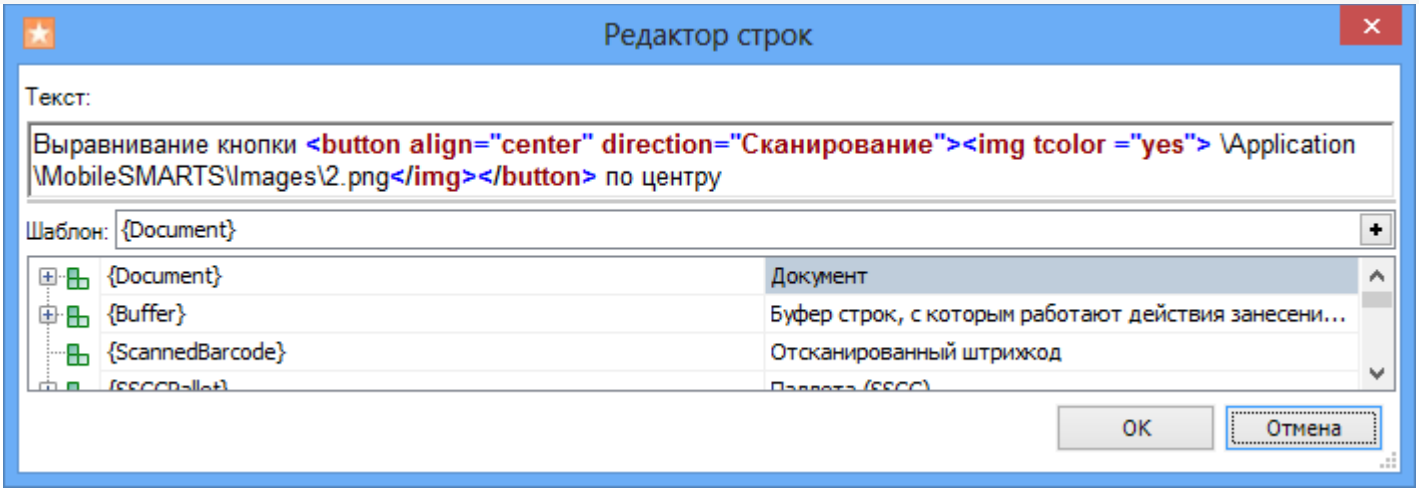

**Для этого в действие «Ввод количества» у свойства «Текст в верхней части окна» нужно добавить кнопку с указанием на действие для перехода и атрибутом выравнивания align =" center " (по центру). Изображение добавляется тегами <img>...</img>, с возможностью форматирования (подробнее про вставку изображения смотрите здесь ).**



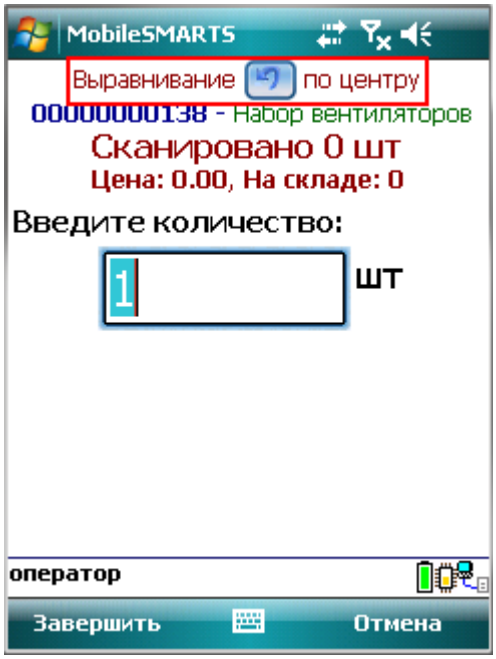

**Наше выражение для добавления кнопки выглядит так:**

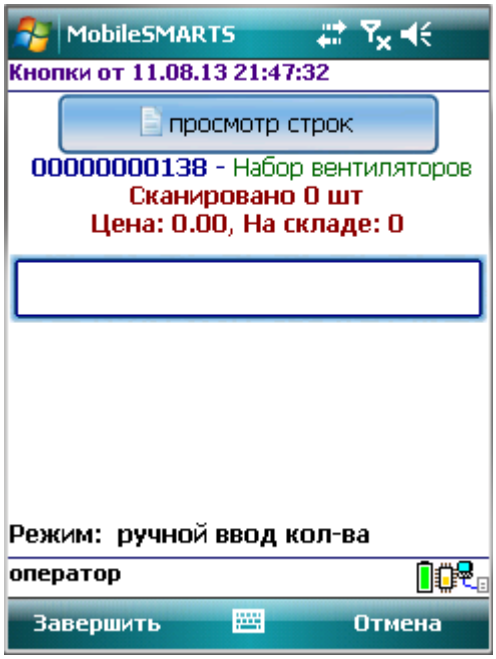
[HTML]

<button align="center" direction="Сканирование"><img tcolor ="yes">
\Application\MobileSMARTS\Images\2.png</img></button>,
где
align =" center " – выравнивание текста по центру, относительно кнопки в строке;
Сканирование – имя действия, на который будет выполнен переход по нажатию кнопки;
<img tcolor ="yes"> \Application\MobileSMARTS\Images\2.png</img> – вставленная картинка с указанием пути до нее на терминале и форматированием (прозрачность цвета фона).



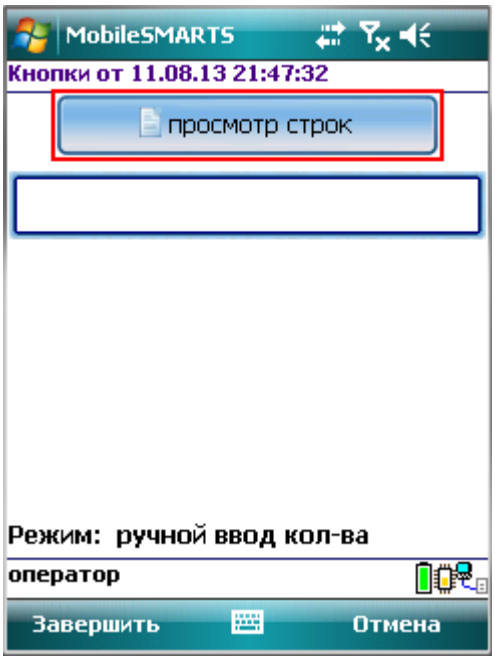

Результатом нажатия кнопки будет переход на действие «Сканирование».



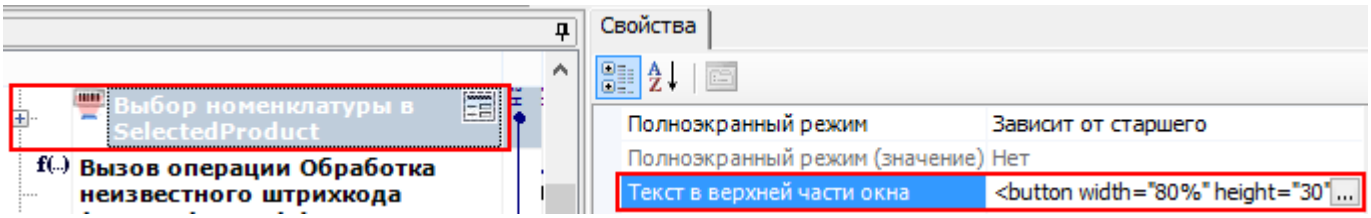




## Пример 3 | Кнопка задана в верхней части окна текстом с изображением

Рассмотрим, как сделать кнопку с заданными размерами в верхней части окна сканирования с текстом и изображением.



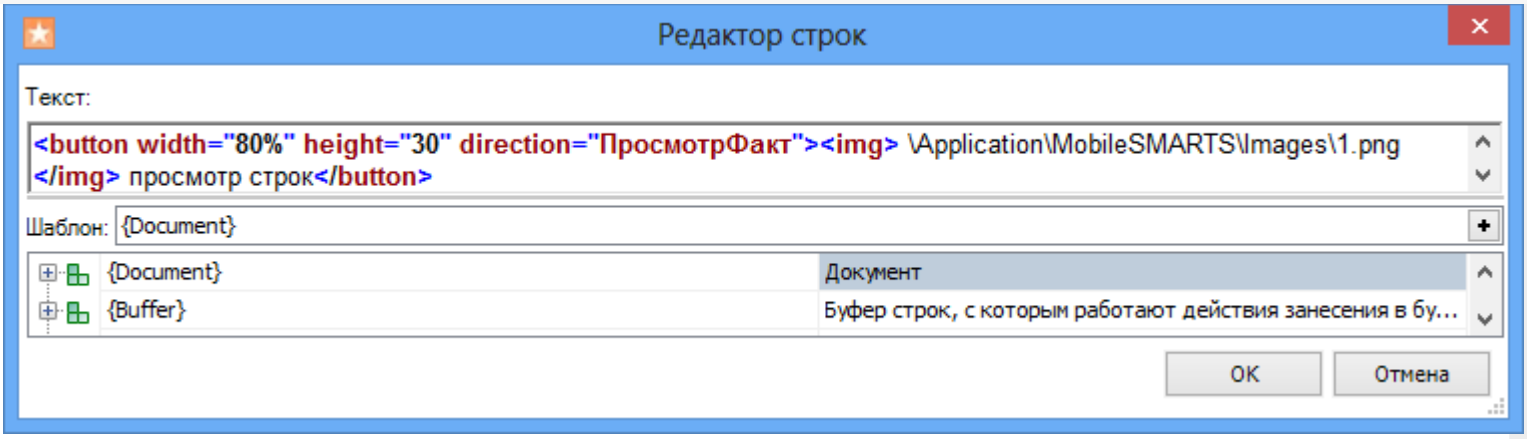

**Для этого в действие «Выбор номенклатуры» у свойства «Текст в верхней части окна» нужно добавить кнопку с указанием на действие для перехода. Изображение добавляется тегами <img>...</img>, с возможностью форматирования (подробнее про вставку изображения смотрите здесь).**



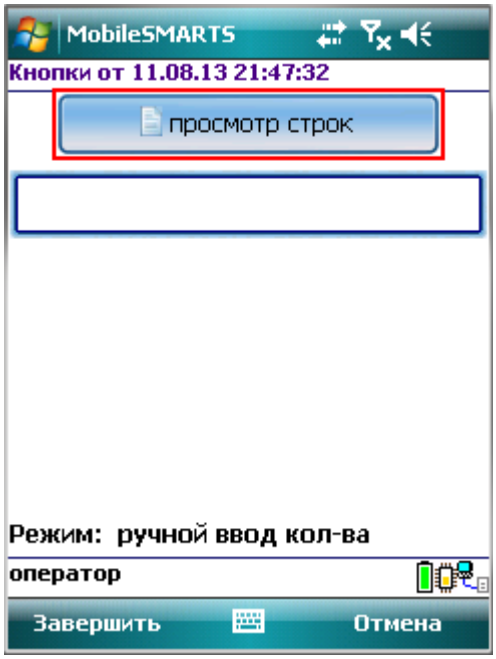

**Наше выражение для добавления кнопки в текст выглядит так:**

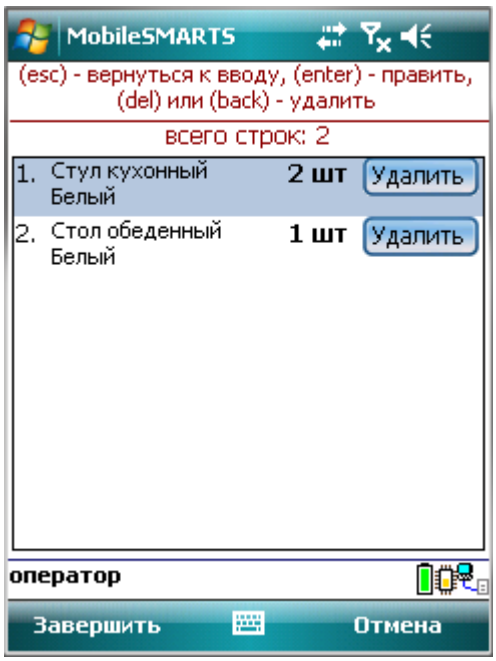
[HTML]

```
<button width="80%" height="30" direction="ПросмотрФакт">
<img>\Application\MobileSMARTS\Images\1.png</img> просмотр строк</button>,
где
width="80%" – ширина кнопки в процентах;
height="30" – высота кнопки в пикселях;
ПросмотрФакт – имя действия, на который будет выполнен переход по нажатию кнопки;
< img >\ Application \ MobileSMARTS \ Images \1. png </ img > – вставленная картинка с
указанием пути до нее на терминале;
просмотр строк – надпись на кнопке.
```




Результатом нажатия кнопки будет переход на действие «ПросмотрФакт» (просмотр строк фактически отсканированных позиций).

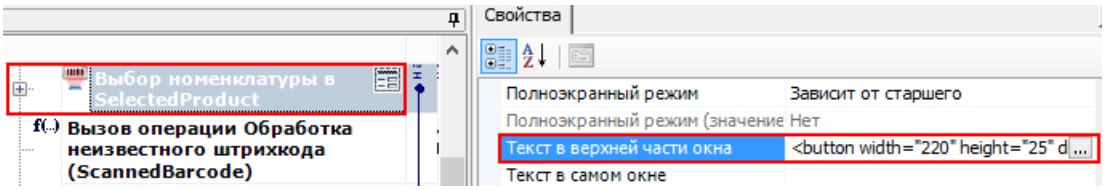



## Пример 4 | Кнопка задана с атрибутом enabled

Рассмотрим, как сделать кнопку, которая будет активна или нет при задании определенных условий.

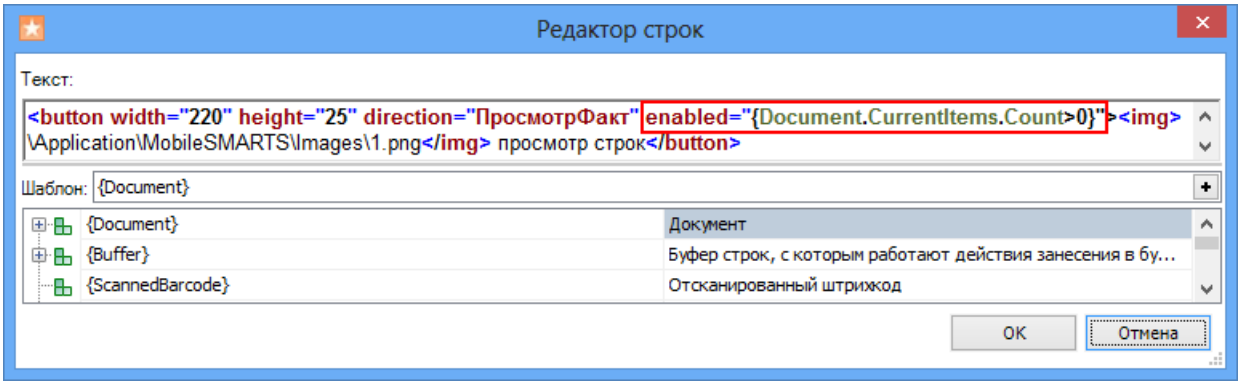
Для примера добавим такую кнопку в верхнюю часть окна в действии «Выбор номенклатуры».



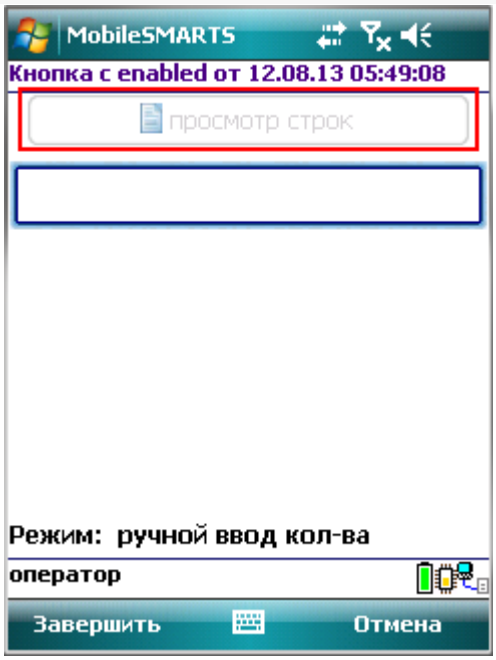

**Добавим атрибут «enabled» и напишем условие, при выполнении которого кнопка будет активна.**

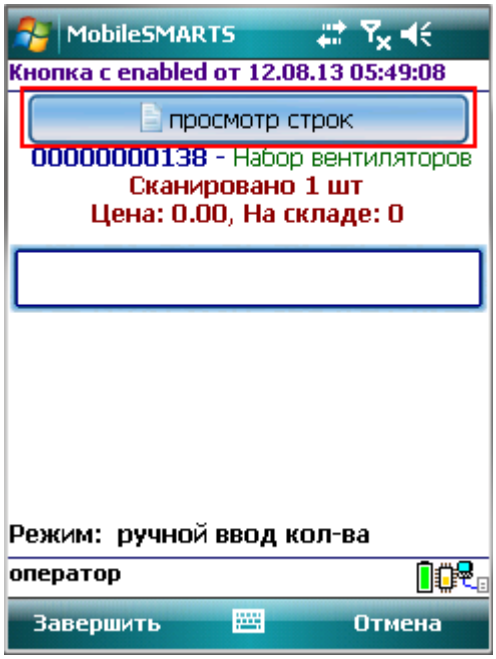
[HTML]

<button width="220" height="25" direction="ПросмотрФакт" enabled="
{Document.CurrentItems.Count>0}"><img> \Application\MobileSMARTS\Images\1.png</img>
просмотр строк</button>,
где
width="220" – ширина кнопки в пикселях;
height="25" – высота кнопки в пикселях;
ПросмотрФакт – имя действия, на который будет выполнен переход по нажатию кнопки;
enabled ="{ Document . CurrentItems . Count >0}" – условие, при выполнении которого кнопка будет активна;
< img >\ Application \ MobileSMARTS \ Images \1. png </ img > – вставленная картинка с указанием пути до нее на терминале;
просмотр строк – надпись на кнопке.




**Результатом будет активная кнопка, при выполнении введенного условия, в нашем случае, если в документе количество строк с набранным товаром больше нуля (когда сканировался хоть один товар). Кнопка будет неактивна, например, когда в новом документе сканирование еще не проводилось, и нет еще ни одной строки с набранным товаром или строки были удалены.**

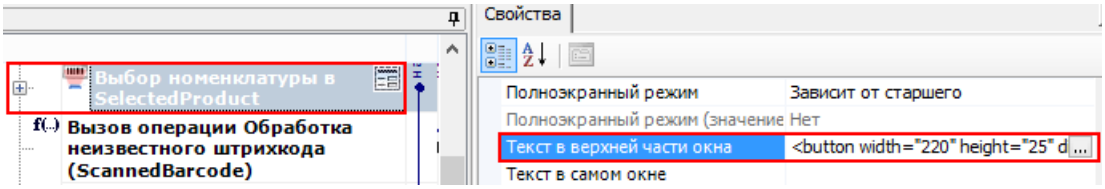



## Пример 5 | Кнопка задана с атрибутом visible

Рассмотрим, как сделать кнопку, которая будет появляться только в случае выполнения определенных условий и кнопки не будет, если это условие не выполняется.

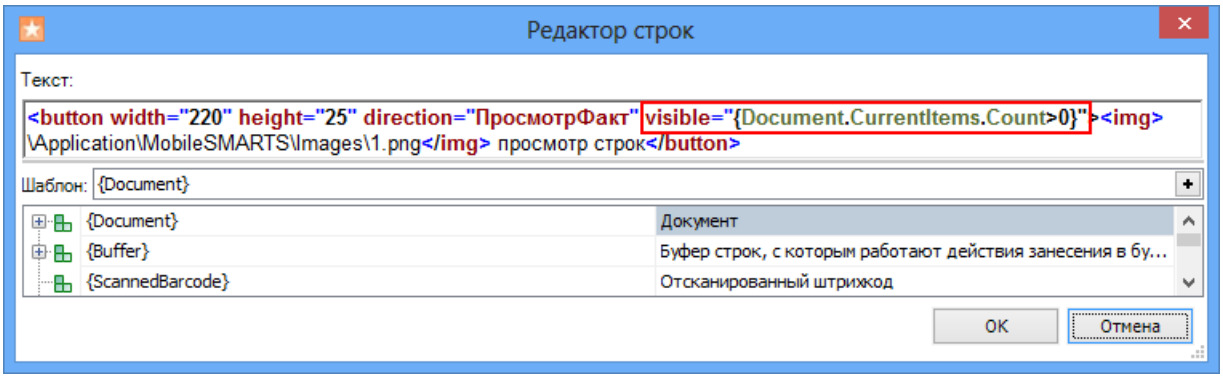
**Для примера добавим такую кнопку в верхнюю часть окна в действии «Выбор номенклатуры».**



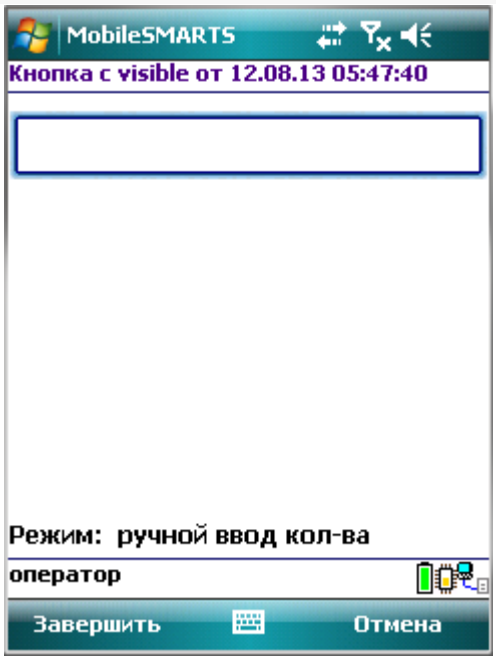

**Добавим атрибут «visible» и напишем условие, при выполнении которого кнопка будет отображаться.**

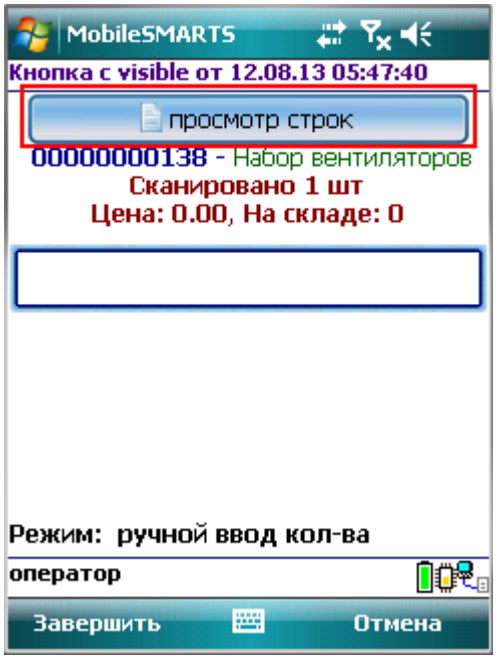
[HTML]

<button width="220" height="25" direction="ПросмотрФакт" visible =" {Document.CurrentItems.Count>0}"><img> \Application\MobileSMARTS\Images\1.png</img> просмотр строк</button>,
где
width="220" – ширина кнопки в пикселях;
height="25" – высота кнопки в пикселях;
ПросмотрФакт – имя действия, на который будет выполнен переход по нажатию кнопки;
visible ="{Document.CurrentItems.Count>0}" – условие, при выполнении которого кнопка будет отображаться;
< img >\ Application \ MobileSMARTS \ Images \1. png </ img > – вставленная картинка с указанием пути до нее на терминале;
просмотр строк – надпись на кнопке.




**Результатом будет отображена кнопка, при выполнении введенного условия, в нашем случае, если в документе количество строк с набранным товаром больше нуля (когда сканировался хоть один товар). Кнопка не будет отображаться, например, когда в новом документе сканирование еще не проводилось, и нет еще ни одной строки с набранным товаром или строки были удалены.**




## Не нашли что искали?

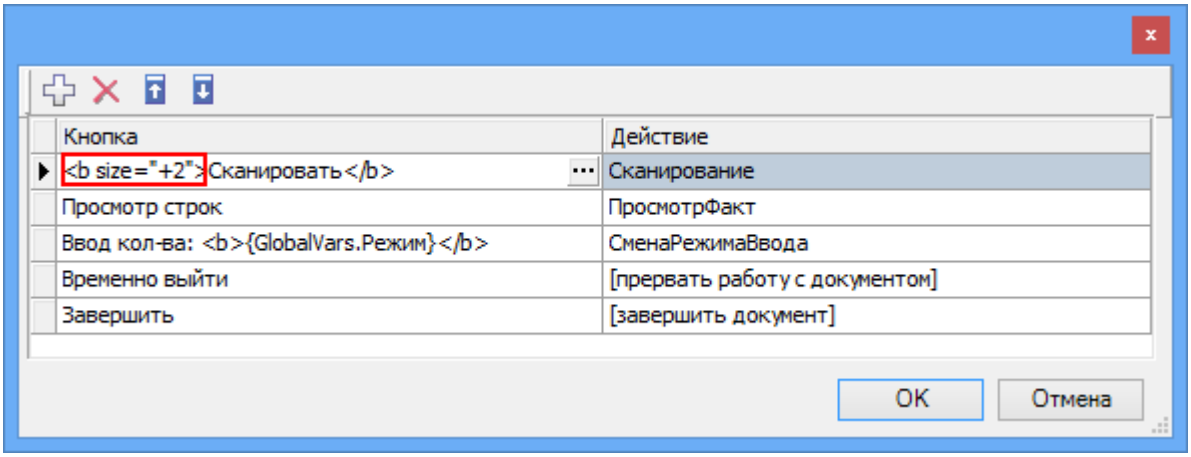
Задать вопрос в техническую поддержку

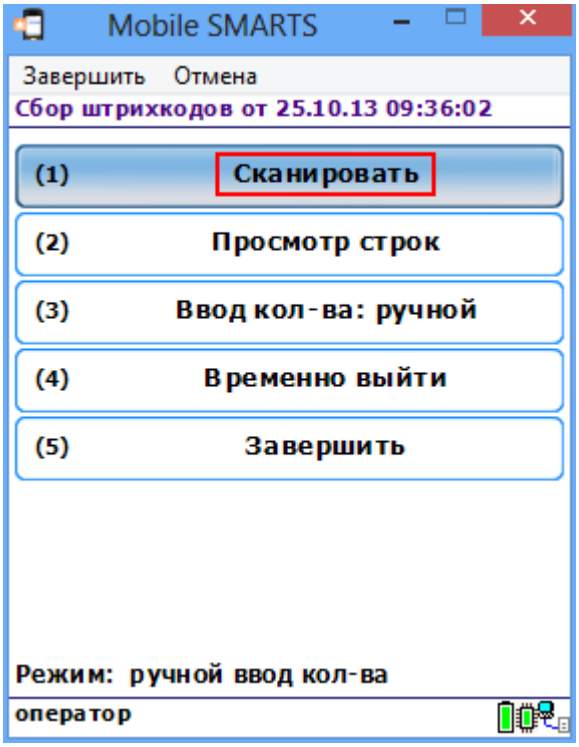
# Примеры изменения относительного размера шрифта в приложении Mobile SMARTS

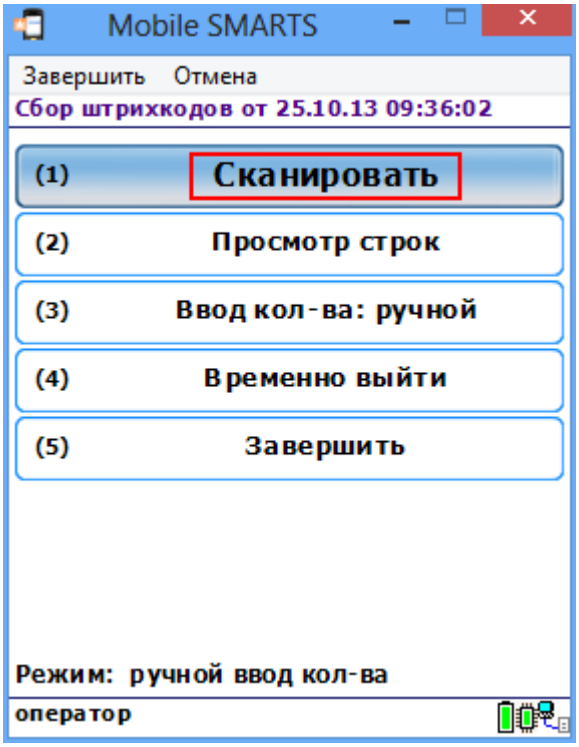
Последние изменения: 2024-03-26

## В кнопках

Для изменения относительного размера шрифта задаем в редакторе строк нужный размер (например, size="+2" –увеличивает шрифт на 2 размера, , size="-1" – уменьшает на 1 размер).

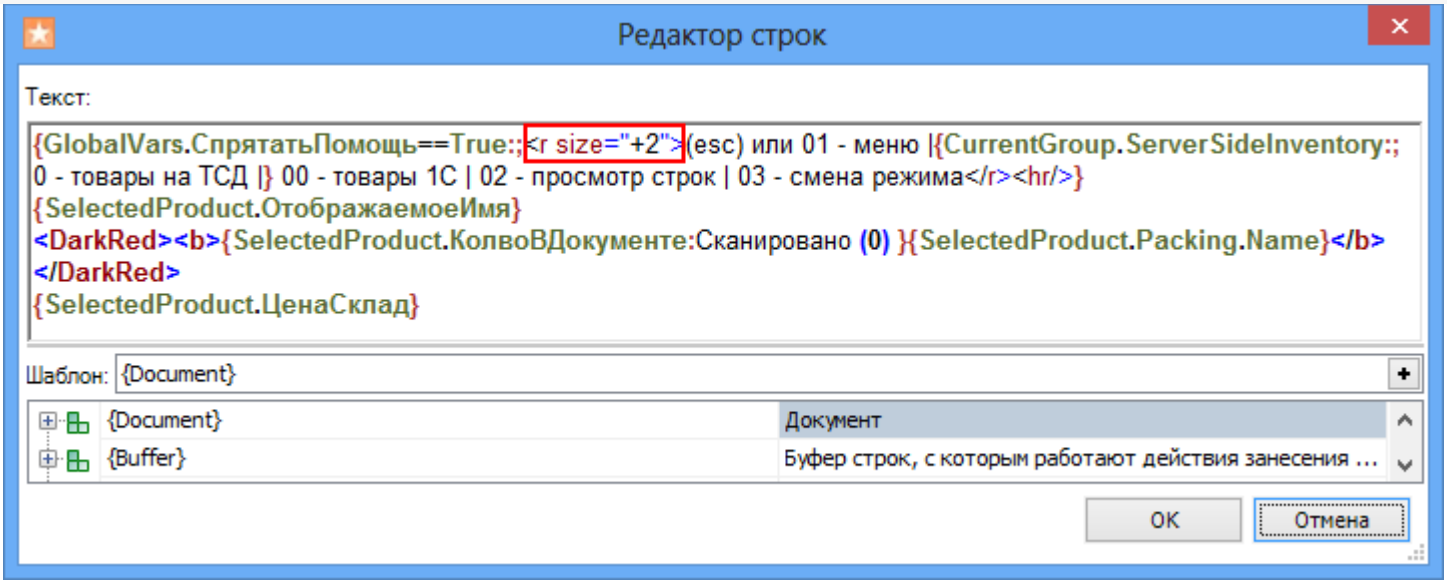


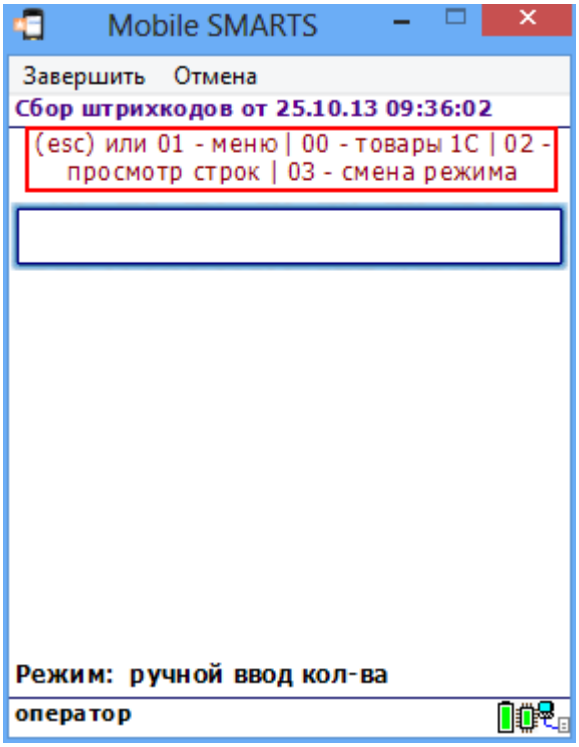



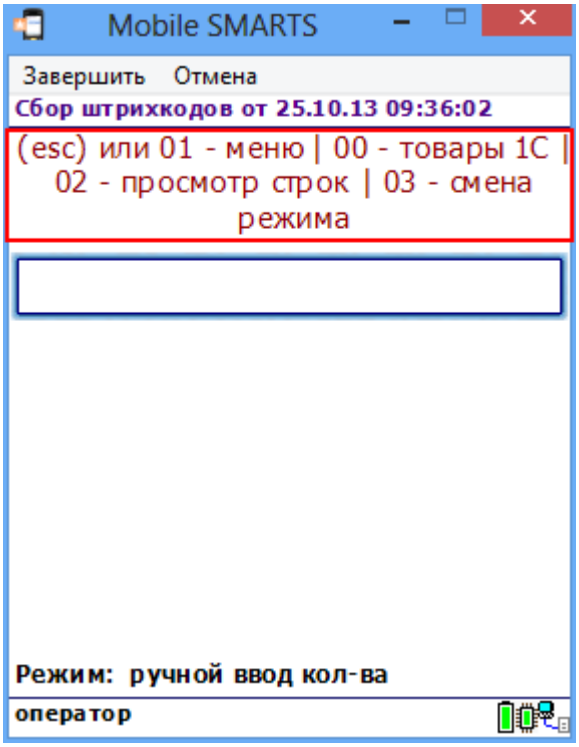




## В тексте

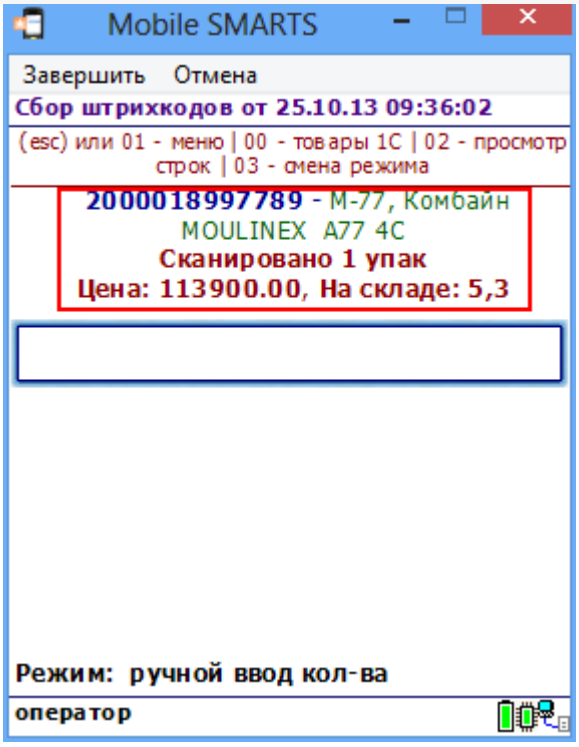
Для изменения относительного размера шрифта задаем в редакторе строк нужный размер (например, size="+2").



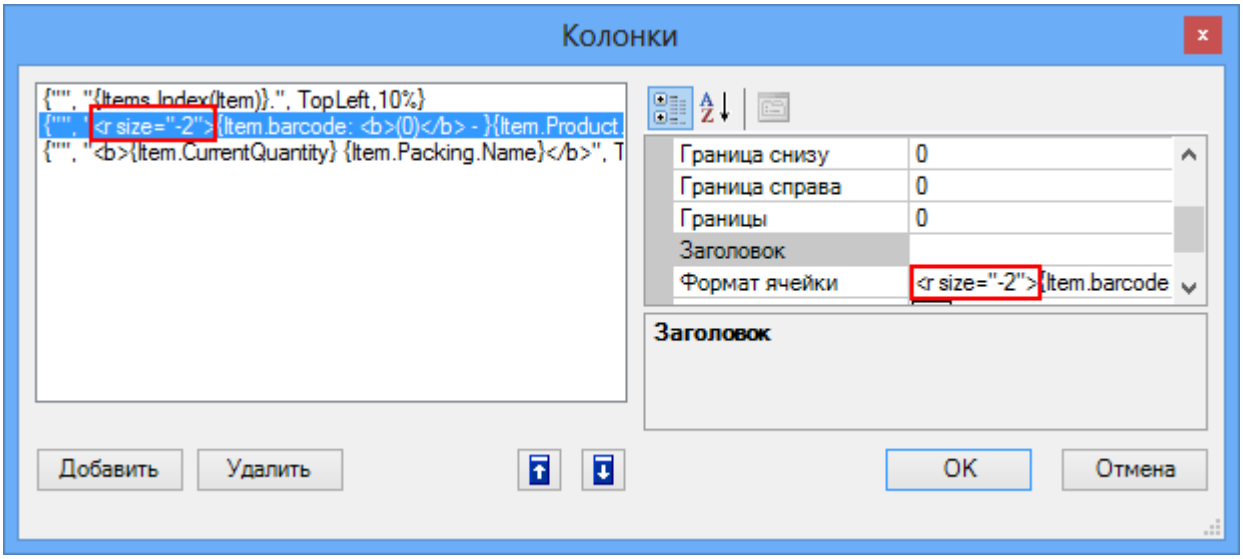



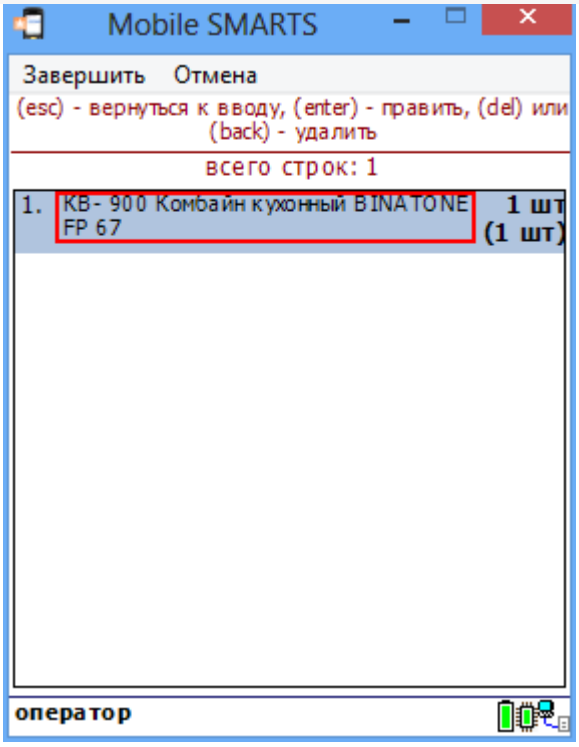

По умолчанию отсканированная позиция отображается с размером шрифта size="+2". Например, если в стиле "NormalFont" установлен размер size="9", то отсканированная позиция будет отображаться размером size="11".

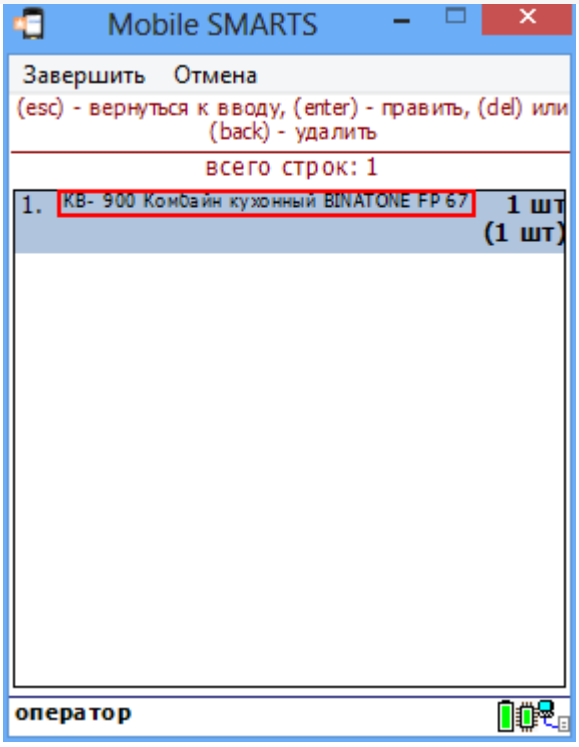



## В тексте списка

Для изменения относительного размера шрифта задаем в редакторе нужный размер (например, size="-2").






форматирование

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Запросы в вычислимых выражениях в платформе Mobile SMARTS

Последние изменения: 2024-03-26

Для отбора строки или нескольких строк из табличной части документа Mobile SMARTS, а также из дополнительной таблицы кроме использования действий Выбор строки и Выбор строк можно использовать выражения запросов. Запросы можно применять в любых местах конфигурации Mobile SMARTS, где используются вычислимые выражения, например, действие «Присваивание», вычислимые поля в различных объектах: таблицах, документе, строках документа.

Запросы имеют следующий синтаксис:

```
select [first] ( * | Имя_Поля, ... n )
from Имя_Источника [as Псевдоним]
where Выражение_Условия
[group by Имя_Поля, ... n]
[sort by Имя_Поля [ asc | desc ], ... n]
```

Здесь:

**first** - выбор первой записи; результатом запроса является одна строка или null. Если оператор **first** не задан, выбираются все строки, удовлетворяющие условию; результатом запроса является коллекция строк. Если не найдено строк, удовлетворяющих условию, коллекция будет пустой.

**( * | Имя_Поля, ... n)** - отбираемые поля. В скобках должна находиться или звездочка (*) - все поля, или список полей через запятую (Имя_Поля, ... n) - отбор только указанных полей.

**Имя_Источника** - имя объекта, из которого выбираются данные. Например, имя дополнительной таблицы, как оно задано в конфигурации Mobile SMARTS. Может быть задан псевдоним с помощью as Псевдоним. Если псевдоним задан, в выражении условия для обращения к объекту-источнику данных должен использоваться заданный псевдоним. Если псевдоним не задан, используется Item (обозначает текущую строку, на которую накладывается условие). Задание псевдонима необходимо, если запрос применяется в вычислимом поле некоторого объекта, являющегося коллекцией строк (например, дополнительная таблица). В этом случае Item является текущей строкой того объекта, который является владельцем вычислимого поля. Пример см. ниже.

**Выражение_Условия** - выражение, содержащее условие отбора строк. Такие же выражения используются в действиях Выбор строки и Выбор строк. Отличие только в том, что вместо Item для обозначения текущей строки может использоваться указанный псевдоним.

**group by Имя_Поля, ... n** - группировка по указанным полям.

**sort by Имя_Поля [ asc | desc ], ... n** - сортировка по указанным полям, перед именем поля может быть задано asc (сортировка по возрастанию) или desc (сортировка по убыванию). По умолчанию, если не указано ни asc ни desc, используется сортировка по возрастанию.

## Пример

Отберем с помощью запроса розничную цену выбранного товара:

```
select first (*) from Цены where Item.ИдНоменклатуры == SelectedProduct.Product.Id &&
Item.ВидЦены == "Розничная"
```

Цены - дополнительная таблица с ценами, в условии указываем поиск по ИдНоменклатуры и виду цены.

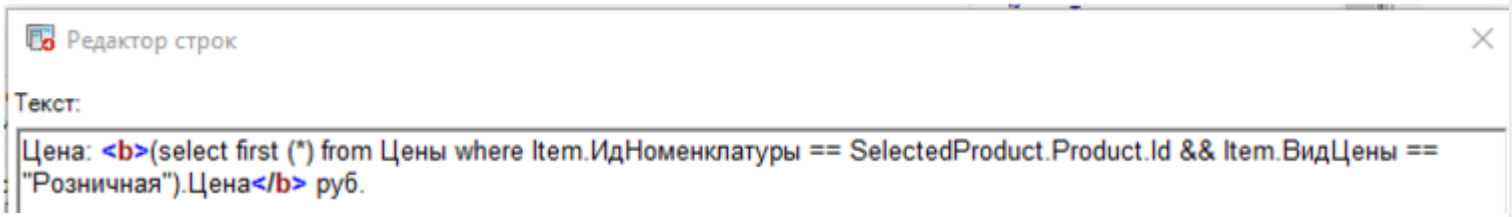
Данное выражение может быть использовано, например, в действии присваивания:

```
СтрокаЦены = select first (*) from Цены where Item.ИдНоменклатуры == SelectedProduct.Product.Id &&
Item.ВидЦены == "Розничная"
```

Если нам нужно получить не строку таблицы, а сразу значение цены, можно сделать так:

```
Цена = (select first (*) from Цены where Item.ИдНоменклатуры == SelectedProduct.Product.Id &&
Item.ВидЦены == "Розничная").Цена
```

Присваивание использовать не обязательно, можно, к примеру, сразу вывести значение цены в заголовке окна:



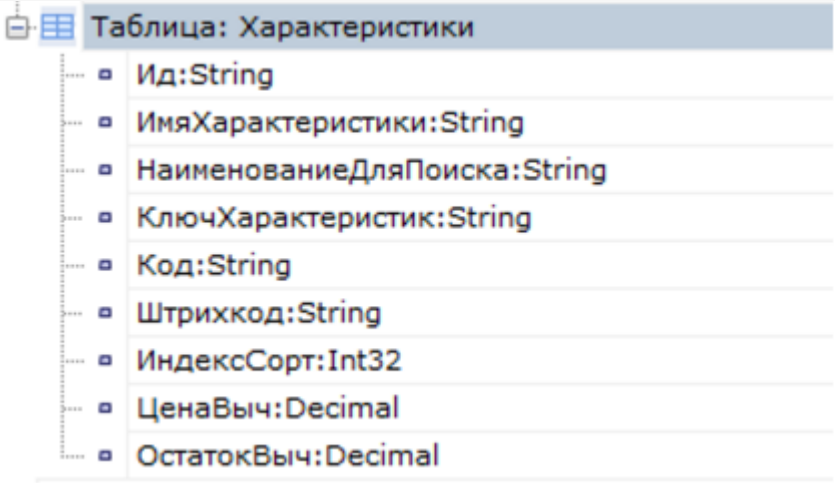

## Пример

Рассмотрим более сложный пример: в таблице Характеристики, содержащей характеристики номенклатуры, создадим вычислимое поле для получения цены из таблицы Цены. Это позволит вывести на экран список характеристик выбранной номенклатуры с их ценами.

Таблица Цены имеет следующие поля:

Цена привязана к номенклатуре, характеристике и упаковке номенклатуры, при этом может быть несколько цен разных видов. Флаг ВидЦеныВнутр равен 1 (заданный по умолчанию вид цен) или 0 (остальные виды цен).

Таблица Характеристики:



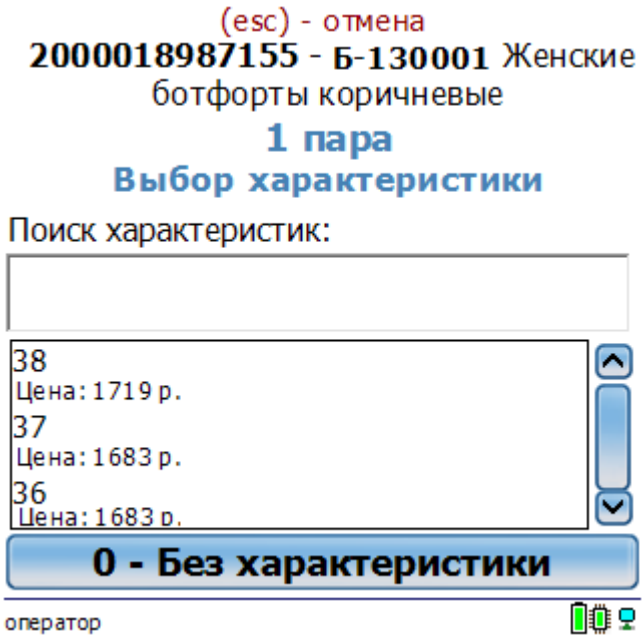

**Для связи с номенклатурой используется поле КлючХарактеристик.**

**Для поля ЦенаВыч задан следующий Шаблон значения:**

```
(select first (*) from Цены as ТаблицаЦен
where ТаблицаЦен.ИдНоменклатуры == SelectedProduct.Product.Id && ТаблицаЦен.Упаковка ==
SelectedProduct.Packing.Id && ТаблицаЦен.Характеристика == Item.ИмяХарактеристики &&
ТаблицаЦен.ВидЦеныВнутр==1).Цена
```

**В выражении условия для обращения к строке таблицы Цены используется заданный псевдоним ТаблицаЦен, Item здесь - указание на текущую строку таблицы Характеристики, в которой определено данное вычислимое поле.**

**Используем поле ЦенаВыч при отображении характеристик в списке выбора:**




Mobile SMARTS, платформа, выражения

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Доработка визуального оформления главного меню приложения на платформе Mobile SMARTS для ОС Android

Последние изменения: 2024-03-26

Экранные формы главного меню приложения относятся к так называемым жестким формам правка которых ранее была недоступна. Но сейчас в последних версиях андроид-клиента для приложений Mobile SMARTS это стало возможно.

Чаще всего для внесения изменений во внешний вид приложения потребуется добавить строки кода (см. примеры ниже) в файл global.css, который находится на ПК в папке по пути «C:\ProgramData\Cleverence\Databases\имя вашей базы\Documents» (путь указан по умолчанию). На ТСД этот файл дублируется, и может иметь названия global.css, global.android.css, global.cf.css.

Рассмотрим конкретнее на примере «Магазина 15», что можно изменить во внешнем виде приложения.

1. Стандартный фон шапки и цвет названия приложения.

Потребуется изменить код в файле global.css (global.android.css, global.cf.css):

Было

Стало

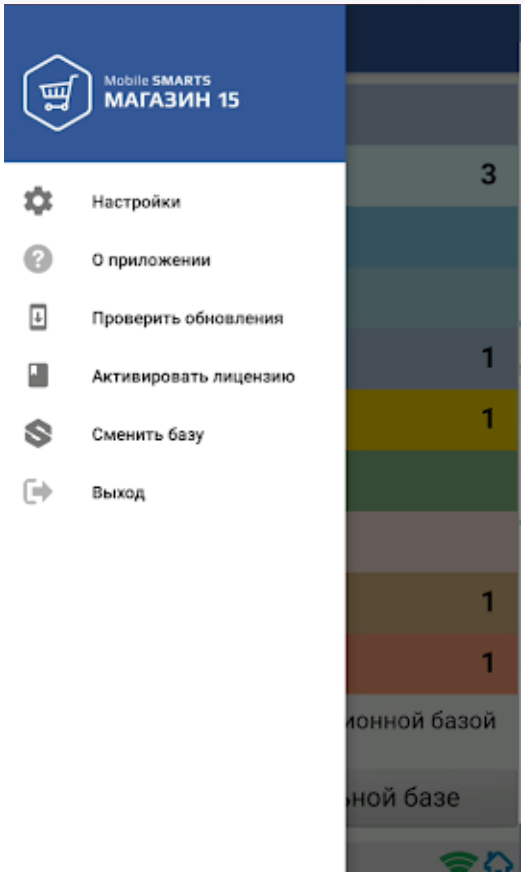
[CSS]

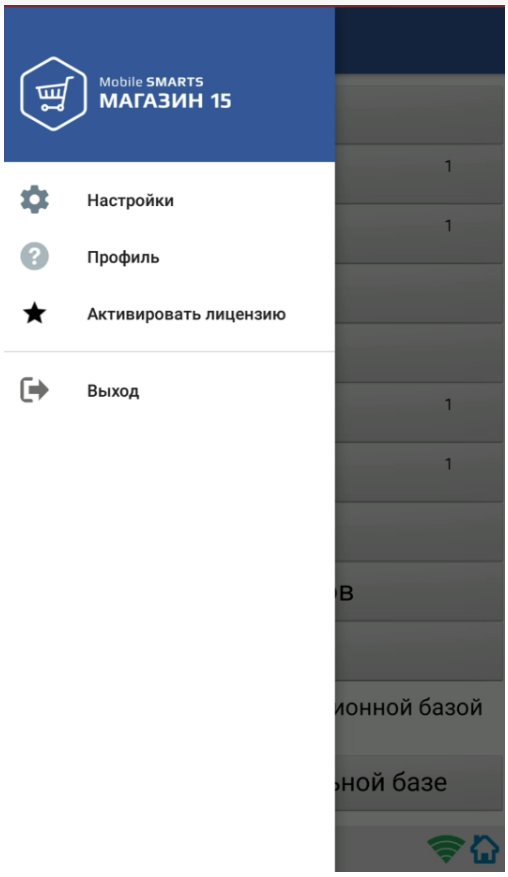
```
{
colorTitle:#EBEBEB;
colorSubTitle:#EBEBEB;
colorHeader:#993535;
colorStatusBar:#7f2a2a;
iconColor:white;
submenuiconcolor:#808080;
}
```

[CSS]

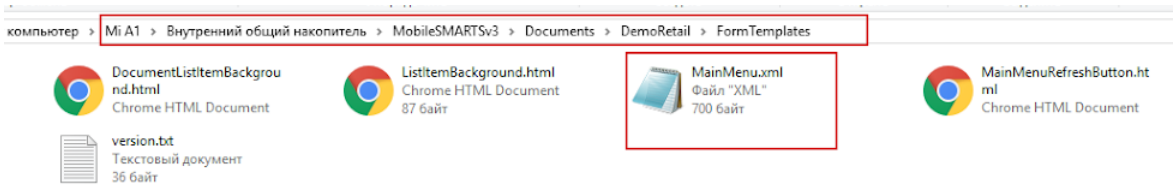
```
.__main_action_bar
{
colorTitle:#928907;
colorSubTitle:#EBEBEB;
colorHeader:#355899;
colorStatusBar:#7f2a2a;
iconColor:white;
submenuiconcolor:#808080;
}
```

2. **Внешний вид «шторки».**




**Для этого потребуется на мобильном устройстве добавить папку FormTemplates в папку нужной вам базы и создать там файл MainMenu.xml, в который записываются следующие строки кода:**




[XML]

```
<?xml version="1.0" encoding="utf-8"?>
<Menu xmlns:clr="http://schemas.cleverence.ru/clr&quot; capacity="16" count="11">
<ChangeFontsAction actionName="Настройки" icon="Set" iconColor="#6A7F8D" />
<DocumentTypeAction actionName="Профиль" documentTypeName="FrontolParams" icon="Q"
iconColor="#6A7F8D" />
<DocumentTypeAction actionName="Активировать лицензию" icon="abc_ic_star_black_16dp"
documentTypeName="#clinternal_ObtainLicense" />
<Separator />
<ChangeUserInfoAction actionName="Сменить пользователя" icon="ic_book_black_24dp"
iconColor="#6A7F8D" />
<ExitAction actionName="Выход" icon="E" iconColor="@null"/
</Menu>
```

3. **Текст на кнопке обмена.**

**Происходит путем добавления в ту же папку FormTemplates файла MainMenuRefreshButton.html со следующим содержимым:**

[HTML]

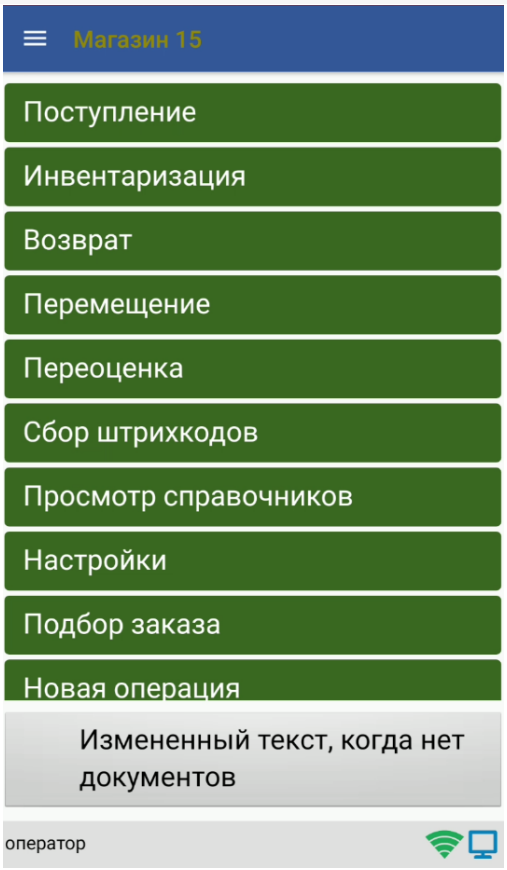
```
<div class="__mainmenu_refresh_button_container">
<div>
<table style="width:100%;">
<tr>
<td width="15%" style="text-align:left; vertical-align:middle;padding-left:20dp;"><img
tcolor="yes" maxwidth="70" maxheight="70">simple.Refresh</img>
</td><td style="text-align:left; vertical-align:middle;"><span class="list_name">
 {__NotSendDocCount>0:Измененный текст, когда есть документы;Измененный текст, когда
нет документов}</span></td>
</tr>
</table>
</div>
</div>
```

4. **Внешний вид кнопок списка операций** меняется путем удаления цветов кнопок, заданных в конфигурации, и добавления в файл global.css (global.android.css, global.cf.css) следующих строк:

[CSS]

```
.__mainmenu_button
{
margin-bottom: 4px;
margin-right: 0px;
margin-left: 0px;
background-color: #3A6721;
corner-radius: 8%;
border: 0px solid #706f6f;
color: white;
}
.__mainmenu_button:hover
{
margin-bottom: 4px;
margin-right: 0px;
margin-left: 0px;
background-color: #6CBC3E;
corner-radius: 8%;
border: 0px solid #706f6f;
color: white;
}
.__mainmenu_button:active
{
margin-bottom: 4px;
margin-right: 0px;
margin-left: 0px;
background-color: #6CBC3E;
corner-radius: 8%;
border: 0px solid #706f6f;
color: white;
}
.__mainmenu_button:disabled
{
margin-bottom: 4px;
margin-right: 0px;
margin-left: 0px;
background-color: #9E9E9E;
corner-radius: 8%;
border: 0px solid #9E9E9E; color: white;
```

**Итог:**




После того как вы внесли правки в файл global.css на ПК, потребуется удалить из папки «MobileSMARTSv3 --> Documents» на ТСД папку базы, для которой вы вносили изменения, и заново подключиться к ней. В противном случае внесенные правки не будут применены к приложению на ТСД.

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Правила именования объектов в Mobile SMARTS

Последние изменения: 2024-03-26

Имена объектов могут содержать русские и латинские заглавные и строчные буквы, цифры, разделенные пробелами (не всегда) или без них.

При написании имени запрещено использовать арифметические знаки и другие специальные символы, которые используются при написании переменных и шаблонов в Mobile SMARTS (например, + - [] {} () _ ! | ; . & =).

| Где используется | Что используется | Примеры наименования | Примеры **неправильного** наименования |
|---|---|---|---|
| Имя типа документа | | | |
| Русские и латинские заглавные и строчные буквы, цифры. Пробелы разрешены. | | | |
| Приход на склад<br><br>1Спереоценка<br><br>Перемещение по ячейкам | | | |
| 1С [переоценка] | | | |
| Имя операции | | | |
| Русские и латинские заглавные и строчные буквы, цифры. Пробелы разрешены. | | | |
| Основной процесс<br><br>Выбор неизвестного товара<br><br>Просмотр ПланФакт | | | |
| Просмотр (ПланФакт) | | | |

| |
|---|
| Имя действия в алгоритме |
| Русские и латинские заглавные и строчные буквы, цифры. Пробелы разрешены. |
| ПрисвоитьШК<br><br>2<br><br>Проверка повтора серийного номера |
| Присвоить_ШК |
| Имя дополнительного поля строки или шапки документа |
| Русские и латинские заглавные и строчные буквы, цифры . Без пробелов. |
| ОтображаемоеИмя<br><br>Serial<br><br>Sn<br><br>ЦенаСклад |
| Отображаемое Имя<br><br>Цена(Склад) |
| Имя дополнительной таблицы |
| Русские и латинские заглавные и строчные буквы, цифры . Без пробелов. |
| НоваяТаблица1 |
| Таблица 1 |

| |
|---|
| Имя колонки в дополнительной таблице |
| Русские и латинские заглавные и строчные буквы, цифры. Без пробелов. |
| Колво |
| _колво |
| Имя переменной в алгоритме |
| Русские и латинские заглавные и строчные буквы, цифры. Без пробелов.<br><br>Имя переменной должно начинаться с буквы или со знака «/». Если имя начинается со слэша («/»), его обязательно нужно экранировать дополнительным слэшем. |
| ИзмененныеСтрокиТовара<br><br>СтрокаССерийнымНомером<br><br>SelectedLine<br><br>d39<br><br>\\\\Images\\All |
| 23<br><br>Selected-Line<br><br>Измененные Строки<br><br>\\Images\All |

объекты

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Форматы вывода данных в Mobile SMARTS

Последние изменения: 2024-03-26

> Перечисленные форматы вывода могут также использоваться и для ввода данных (например, в шаблонах штрихкодов), кроме ввода даты/ времени (подробнее).

## Форматы для вывода чисел

В таблице приведены только наиболее практичные форматы из спецификации .NET (http://msdn.microsoft.com/ru-ru/library/dwhawy9k.aspx и http://msdn.microsoft.com/ru-ru/library/0c899ak8.aspx) плюс специальные форматы, существующие только в Mobile SMARTS. Некоторые форматы используют числовые параметры — в тех местах, где можно вставить число, в таблице ниже используется *.

| Формат | Описание |
|---|---|
| N* или n* | |
| Дробное число с ограничением на число знаков после запятой. Если ограничение не указано, то используется 2 знака после запятой. | |
| {1000:N3} = 1000,000<br><br>{12.519:N} = {12.519:N2} = 12,52<br><br>{7:N} = {7:N2} = 7,00 | |
| C или c | |
| Сумма в рублях | |
| {100:c} = 100,00р.<br><br>{1200.12:c} = 1 200,12р. | |

Набор из «0#.,»

Фиксированный формат вывода числа.

«0» — обязательная цифра,

«#» — необязательная цифра,

«.» — десятичная точка,

«,» — разделитель числовых разрядов.

Все остальные символы ничего не означают и просто копируются в результат.

Можно задать до трех секций формата, разделенных точкой с запятой. Первая секция — для положительных чисел, вторая — для отрицательных, третья — для ноля.

{2.5:0.00} = 2,50

{2.5:0.0#} = 2,5

{2.527:0.0#} = 2,53

{2.49:0.0;<red>-0.0</red>;ноль} = 2,5

{-2.49:0.0;<red>-0.0</red>;ноль} = -2,5

{0:0.0;<red>-0.0</red>;ноль} = ноль

{17:0.#} = 17

{17.2:0.#} = 17,2

{17.2:000.00} = 017,20

{17.2:###.##} = 17,2

{10000:00,00.00} = 1 00 00,00

W или w

Количество прописью. W — с большой буквы, w — с маленькой

{2:W} = Два

{100.24:w} = сто

{1341:W} = Одна тысяча триста сорок один

| |
|---|
| Wf или wf |
| Количество прописью женского рода. W — с большой буквы, w — с маленькой |
| {2:W} = Две<br><br>{100.24:w} = сто<br><br>{1341:W} = Одна тысяча триста сорок одна |
| WRUR или wRUR |
| Сумма в рублях прописью. Если ноль копеек, то копейки не выводятся. W — с большой буквы, w — с маленькой |
| {2:WRUR} = Два рубля<br><br>{100.247:wRUR} = сто рублей 25 копеек<br><br>{1341:WRUR} = Одна тысяча триста сорок один рубль |
| Wrur или wrur |
| Сумма в рублях прописью с копейками. W — с большой буквы, w — с маленькой |
| {2:Wrur} = Два рубля 00 копеек<br><br>{100.207:wrur} = сто рублей 21 копейка<br><br>{1341:Wrur} = Одна тысяча триста сорок один рубль 00 копеек |
| WUSD, wUSD,Wusd или wusd |
| То же самое, что и с рублями RUR, но для долларов. |
| {2:WUSD} = Два доллара<br><br>{100.207:wUSD} = сто долларов 21 цент<br><br>{1341:Wusd} = Одна тысяча триста сорок один доллар 00 центов |

| |
|---|
| RURc |
| Количество копеек (не округленно, а обрезано) |
| {100.207:RURc} = 20 |
| USDc |
| Количество центов (не округленно, а обрезано) |
| {100.207:USDc} = 20 |

## Форматы для вывода строк

**В спецификации .NET у строк нет форматов вывода, но они есть в Mobile SMARTS. Форматы используют числовые параметры — в тех местах, где можно вставить число, в таблице ниже используется «*» (звездочка).**

| Формат | Описание | П |
|---|---|---|
| T* | | |
| Обрезает строку до * символов | | |
| {"ABCD»:T3} = ABC<br><br>{"ABCD»:T8} = ABCD | | |
| E* | | |
| Обрезает строку до * символов и добавляет троеточие (...) | | |
| {"ABCD»:E3} = ABC...<br><br>{"ABCD»:E8} = ABCD | | |

| |
|---|
| M* |
| Обрезает строку до * символов и добавляет троеточие (...) внутри строки |
| {"ABCDEFGHI»:M7} = AB...HI<br><br>{"ABCDEFGHI»:M8} = ABC...HI<br><br>{"ABCDEFGHI»:M12} = ABCDEFGHI |

## Форматы для вывода дат и времени

**В .NET не существует отдельного типа «дата» и отдельного типа «время». Есть один общий тип «дата и время», поэтому везде, где упоминается дата, на самом деле имеется в виду дата и время, даже для даты документа. В таблице приведены наиболее практичные форматы из спецификации .NET (http://msdn.microsoft.com/ru-ru/library/az4se3k1.aspx и http://msdn.microsoft.com/ru-ru/library/8kb3ddd4.aspx).**

> **Для ввода даты можно использовать только формат {ExpiredDate:yyMMdd}, т.к. при вводе требуется два символа для года.**

| **Формат** | **Описание** |
|---|---|
| d | |
| Короткий формат даты | |
| {дата:d} = 20.04.2009 | |
| D | |
| Длинный формат даты | |
| {дата:D} = Понедельник, 20Апреля 2009 | |
| ddd | |
| День недели | |
| {дата:ddd} = Понедельник | |

| |
|---|
| g |
| Короткий формат даты + короткий времени |
| {дата:g} = 20.04.2009 10:07 |
| t |
| Короткий формат времени |
| {дата:t} = 10:07 |
| T |
| Полный формат времени (с секундами) |
| {дата:T} = 10:07:12 |
| Набор из «mMhHys» |
| Конкретный формат вывода даты и времени, составленный из следующих специальных обозначений и любых других символов: |
| |
| |
| «y» — последняя цифра года,<br><br>«yy» — последние две цифры года,<br><br>«yyy» или «yyyy» — все цифры года,<br><br>«M» — месяц прописью и день,<br><br>«MM» -две цифры месяца,<br><br>«MMM» — месяц прописью сокращенно,<br><br>«MMMM» — месяц прописью,<br><br>«dd» — две цифры дня,<br><br>«ddd» — день прописью сокращенно,<br><br>«dddd» — день прописью, |

| |
|---|
| «m», «mm», «mmm» и т. д. — минуты,<br><br>«s», «ss», «sss» и т. д. — секунды,<br><br>«f», «ff», «fff» и т. д. — миллисекунды,<br><br>Используя эти обозначения можно составлять любые нужные форматы отображения. |
| {дата:y абв} = 9 абв<br><br>{дата:(0:yy)} = 09,<br><br>{дата:yyyy} = 2009<br><br>{дата:M} = апрель 20<br><br>{дата:MM} = 04<br><br>{дата:MMM} = апр<br><br>{дата:MMMM} = Апрель<br><br>{дата:dd} = 20<br><br>{дата:ddd} = Пн<br><br>{дата:ddd} = Понедельник<br><br>{дата:m} = {дата:mm} = 07<br><br>{дата:s} = {дата:ss} = 12<br><br>{дата:f} = 9, {дата:ff} = 91, {дата:fff} = 912<br><br>{дата:MMdd} = 04 19<br><br>{дата:dd-MM-yy} = 19-04-09 |

## Форматы для вывода булевых выражений

**В Mobile SMARTS добавлено форматирование выражений (истина/ложь), чтобы иметь возможность вывести различный текст, в зависимости от значения выражения.**

**Формат выражения:**

**{выражение:вывод для Истина;вывод для Ложь}**

**Выражение — выражение, значением которого является bool;**

**Вывод для Истина — текст, который будет выведен, если значение выражения = True (Истина);**

**Вывод для Ложь — текст, который будет выведен, если значение выражения = False (Ложь);**

| Примеры |
| --- |
| {True:Да;Нет} |
| Будет выведен текст «Да». |
| {False:Да;Нет} |
| Будет выведен текст «Нет». |
| {GlobalVars.Режим == «авто»:автоматич. ввод кол-ва;ручной ввод кол-ва} |
| Если включен авто режим ввода количества, то будет выведена надпись «автоматич. ввод кол-ва», если авто режим отключен, то будет выведено сообщение «ручной ввод кол-ва». |
| {SelectedProduct==null:товар не выбран;товар выбран} |
| Если SelectedProduct еще не присвоен (не выбран), то будет выведено сообщение «товар не выбран». |

**Кроме конкретного текста в качестве формата, можно использовать шаблоны, вместо которых будут подставляться соответствующие значения для вывода, или не вписывать вообще ничего.**

| Примеры |
| --- |
| {SelectedProduct== null :;} |
| Выведено ничего не будет, в любом случае. |
| { РежимОтображенияТекста == False :;{ SelectedProduct .Barcode } { SelectedProduct. Product. Name }} |
| Если отображение текста выключено, то выведено ничего не будет, если отображение текста выключено, то будет выведен штрихкод и наименование товара. |
| {Item.FirstStorage==null:любая;{Item.FirstStorage.Name}}<br><br>в данном случае Item — строка документа |
| Если ячейка в строке документа не указана, то будет выведено «любая», если указана, то будет выведено имя этой ячейки. |

форматы, выражения, шаблоны

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Добавление и форматирование изображений в Mobile SMARTS

Последние изменения: 2024-03-26

Кроме текста с различным форматированием клиент Mobile SMARTS позволяет вставлять изображения.

Для вставки изображений используется тег <img>...</img>.

> <img tcolor="yes,no,цвет" size="normal,stretch" maxwidth="..." maxheight="...">здесь необходимо указать путь к картинке или задать шаблон</img>

| Атрибут | Значение |
|---|---|
| size ="normal" | |
| Изображение вставляется «как есть», без масштабирования. Если места для полного отображения недостаточно, то она будет обрезана. | |
| size="stretch" | |
| Увеличивает (уменьшает) размер картинки до необходимого размера. Возможный размер определяется свойствамиmaxwidth, maxheight или размером области, куда отображается картинка (например, шириной экрана, или размерами ячейки в действии отображения отчета). | |
| tcolor ="yes" | |
| Используется прозрачность цвета фона, которая задается, используя цвет, взятый из первого пикселя картинки с координатами (0,0). | |
| tcolor = "no" | |
| Прозрачность фона не используется. | |
| tcolor ="Blue" | |
| Прозрачность цвета фона задается названием цвета или в виде #XXXXXX. | |

| |
|---|
| maxwidth = "200" |
| Максимальная ширина картинки (задается в пикселях или в % от ширины экрана). |

| |
|---|
| maxheight = "100" |
| Максимальная высота картинки (задается в пикселях или в % от высоты экрана). |
| align="top" |
| Выравнивание текста по верхней границе изображения. |
| align="center" |
| Выравнивание текста по центру изображения. |
| align="bottom" |
| Выравнивание текста по нижней границе изображения. Задан по умолчанию. |

Само изображение для отображения может быть задано несколькими путями:

| **Шаблон** |
|---|
| **Путь к файлу** |
| **Описание** |
| <img> \Images\picture.jpg</img><br><br><img> \Flash\Images\picture.jpg</img> |
| \Images\picture.jpg<br><br>\Flash\Images\picture.jpg |
| Изображение находится по заданному абсолютному пути на терминале. |
| <img>picture.jpg</img><br><br><img>Images\picture.jpg</img> |
| \Application\MobileSMARTS\picture.jpg<br><br>\ Application\MobilesSMARTS\Images\picture.jpg<br><br>Программа установлена в папку \Application\MobileSMARTS |
| Изображение ищется по пути <Папка программы на терминале>\<заданный относительный путь> |

| |
|---|
| <img>{ПеременнаяСПутем}</img> |
| \Flash\Images\pic1.jpg<br>Переменная в сессии {ПеременнаяСПутем}="\Flash\Images\pic1.jpg" |
| Изображение ищется по пути, лежащему в переменной сессии. |
| <img>{ПеременнаяСОбъектом}</img> |
| Нет пути, отображается картинки из памяти. |
| Отображается объект картинки, лежащий в памяти терминала в переменной {ПеременнаяСОбъектом}. Такой объект может быть получен, например, запросом в учетную систему, и не требует физического сохранения, а может быть просто отображен из переменной, в которой он хранится. |

интерфейс, изображения, форматирование

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Примеры вставки и форматирования изображений в Mobile SMARTS

Последние изменения: 2024-03-26

## Пути к изображению

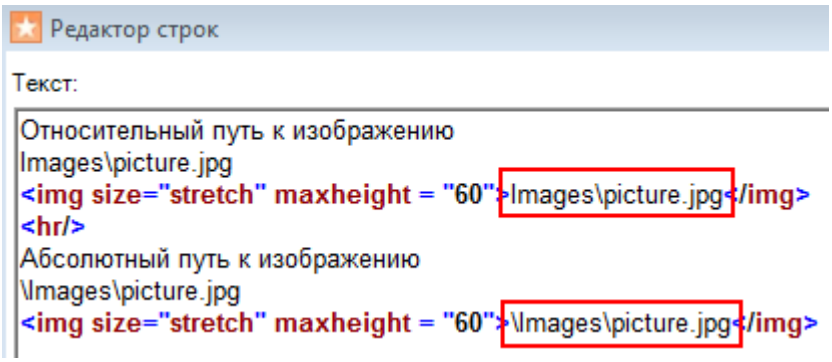
**Пример 1 | Относительный и абсолютный пути к изображению**

**Путь к изображению может быть относительный и абсолютный.**

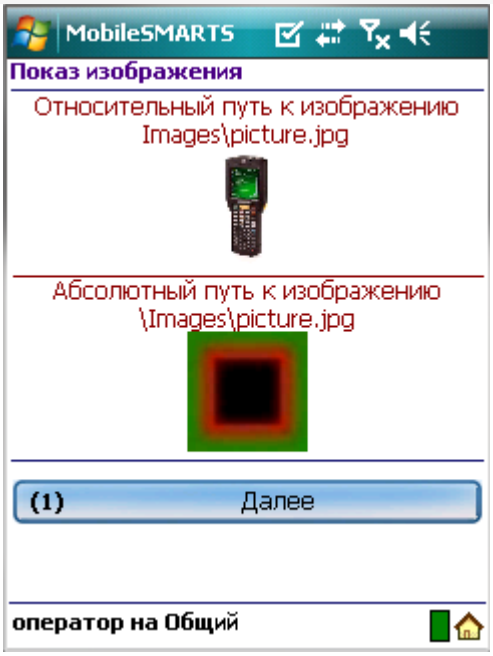
**Возьмем два разных изображения, но с одинаковым названием (picture.jpg), лежащих в разных папках на ТСД. Первое в папке Images установки программы на ТСД, второе в папке Images на флеш накопителе ТСД.**

**Зададим относительный и абсолютные пути к изображениям.**

```
Относительный путь к изображению
Images\picture.jpg
<img size="stretch" max>Images\picture.jpg</img>
<hr/>
Абсолютный путь к изображению
\Images\picture.jpg
<img size="stretch" max>\Images\picture.jpg</img>
```



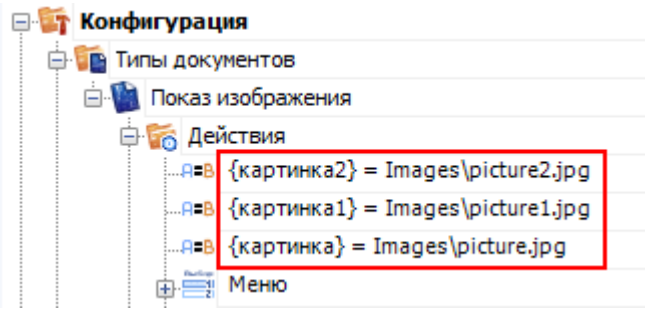

**Результат на терминале:**




Первое изображение находилось в папке Images установки программы на ТСД (относительный путь), второе в папке Images на флеш накопителе ТСД (абсолютный путь – любое место на ТСД).

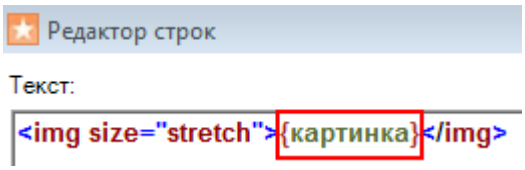
**Пример 2 | Изображение задано переменной**

Чтобы вставлять изображения в виде переменных сначала необходимо их создать. Переменные имеют вид {любое название}=путь к изображению.




После того, как переменная создана, можно вместо пути указывать нужную переменную.

```
< img size="stretch">{картинка}</img>
```

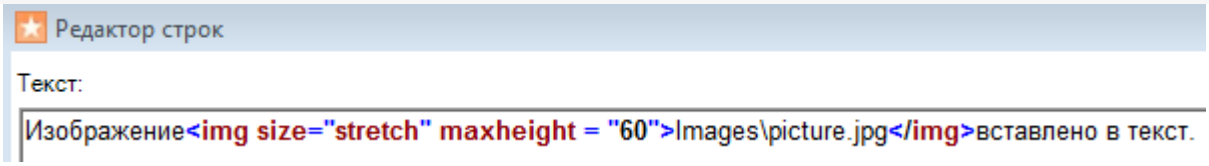



## Вставка изображения в текст

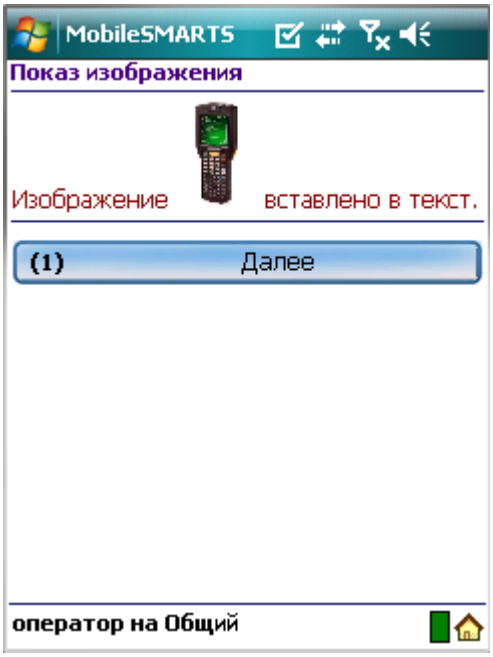
**Пример 1 | Вставка изображения в текст**

Существует возможность вставлять изображения в текст. Для этого тег <img>...</img> необходимо размещать в нужном месте текста.

```
Изображение <img size="stretch" max>Images\picture.jpg</img> вставлено в текст.
```



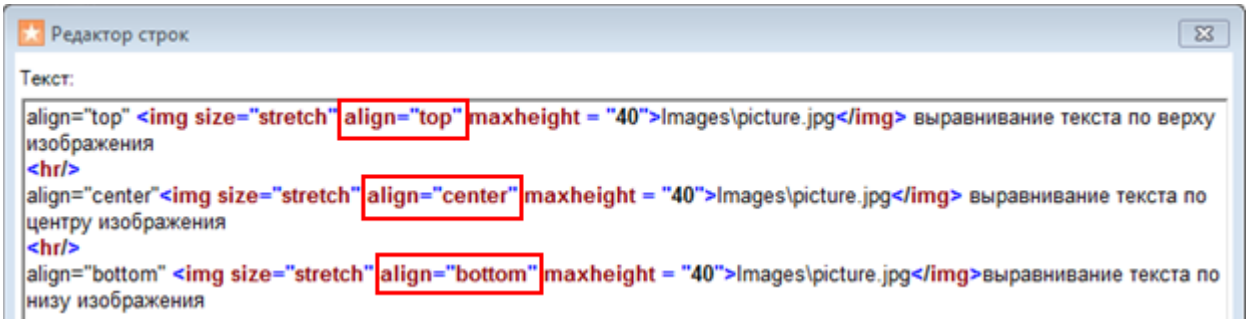

**Результат на терминале:**




**Пример 2| Выравнивание текста по отношению к изображению**

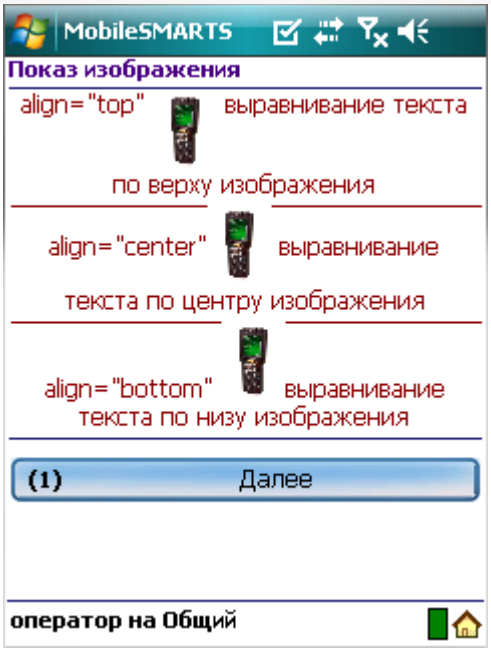
**При вставке изображения в текст, можно задавать выравнивание текста по отношению к изображению (по верху, по центру, по низу). Для этого необходимо указать одно из значений выравнивания.**

```
<img size="stretch" align="top" max>Images\picture.jpg</img> выравнивание текста по верхней границе изображения

 <img size="stretch" align="center" max>Images\picture.jpg</img> выравнивание текста по центру изображения

 <img size="stretch" align="bottom" max>Images\picture.jpg</img> выравнивание текста по нижней границе изображения
```




> Если атрибут «align» не прописан, происходит выравнивание по нижнему краю изображения.

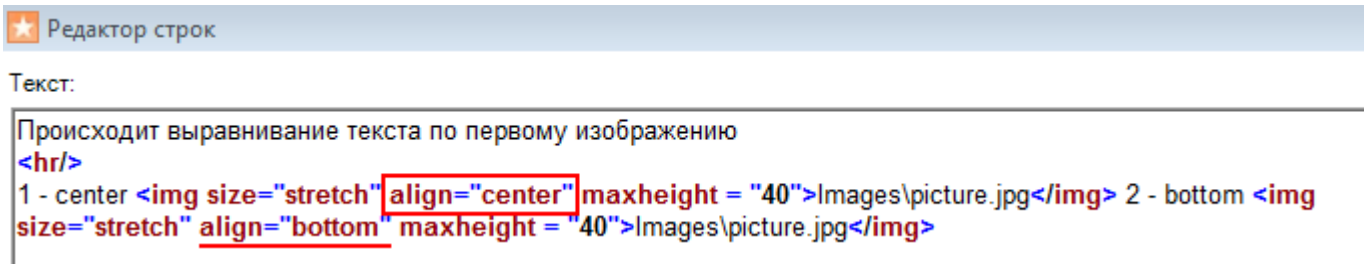
**Результат на терминале:**



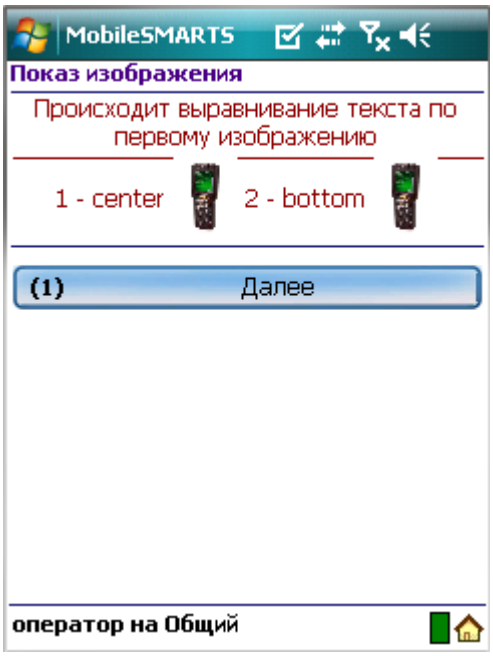

**Пример 3| Несколько изображений на одной строке с разными выравниваниями**

**Если на одной строчке вставлено несколько изображений с разными выравниваниями, то весь текст выравнивается по значению заданному у первого изображения.**

```
1 - center <img size="stretch" align="center" max>Images\picture.jpg</img> 2 - bottom <img size="stretch" align="bottom" max>Images\picture.jpg</img>
```




**Результат на терминале:**



## Вставка изображения в кнопку меню

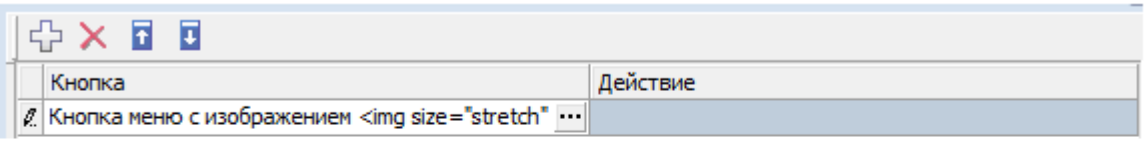
**Пример 1 | Вставка изображения в кнопку меню с ограничением размеров**

Для вставки изображения в кнопку меню тег <img>...</img> необходимо размещать в самой кнопке.

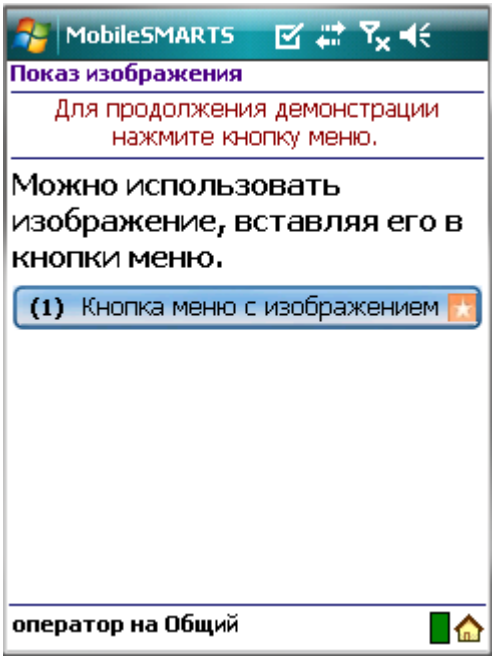
```
<img size="stretch" max max>{картинка1}</img>
```

В данном примере, мы указали размер области, в которую вставляем изображение, задав максимальную ширину и высоту. Дополнительно задали атрибут «size="stretch"», который увеличит (уменьшит) изображение до этих размеров. Само изображение задано переменной.

Не обязательно указывать оба параметра размеров, можно указать только один (высоту или ширину изображения).



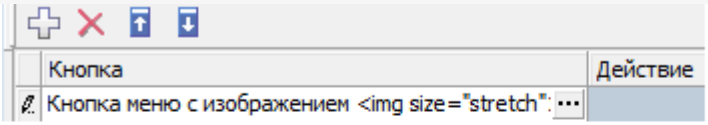
Результат на терминале:

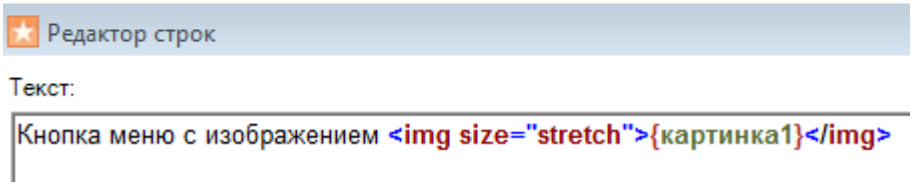



**Пример 2 | Вставка изображения в кнопку меню без конкретных заданных размеров**

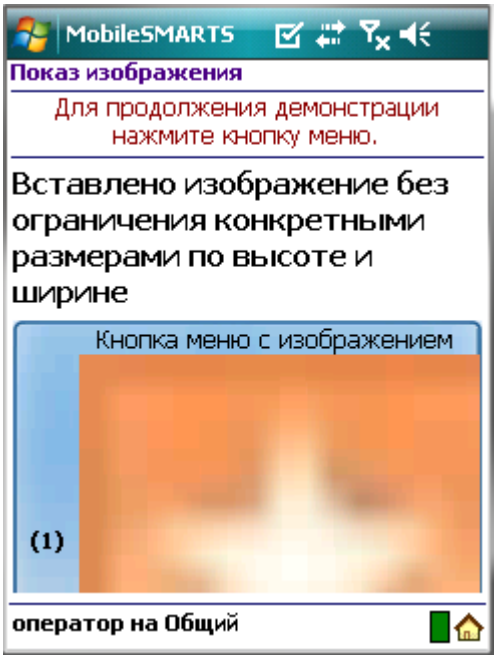
Посмотрим, что будет, если не задавать конкретные размеры области изображения.

```
<img size="stretch">{картинка1}</img>
```




**Результат на терминале:**




**Изображение растянулось до максимально возможного размера.**

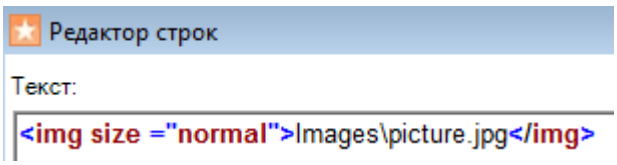
> **Рекомендуется задавать конкретные размеры области изображения для вставки в кнопку меню, для корректного отображения картинки.**

## Размеры изображения

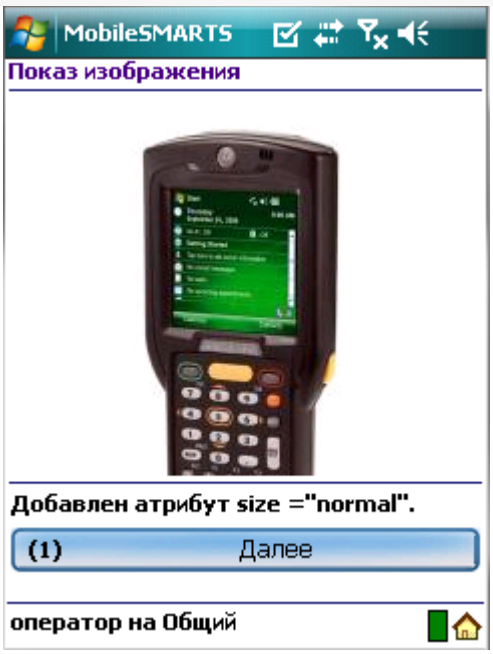
**Пример 1 | Размер без масштабирования**

**Чтобы вставить изображение без масштабирования необходимо задать атрибут size ="normal".**

```
<img size ="normal">Images\picture.jpg</img>
```




**Результат на терминале:**

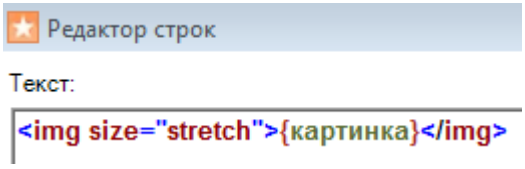



Изображение было обрезано, т.к. полностью не уместилось на экране.

**Пример 2 | Размер с масштабированием**

Чтобы увеличить (уменьшить) размер изображения до размера области вставки, необходимо задать атрибут size ="stretch".

```
<img size="stretch">{картинка}</img>
```



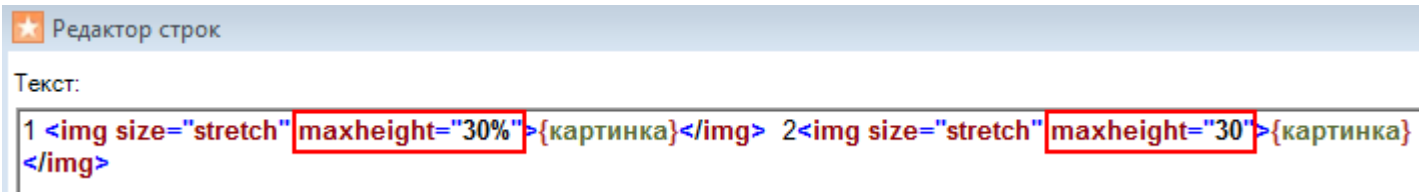

Результат на терминале:

**Пример 3 | Задание конкретных размеров изображения**

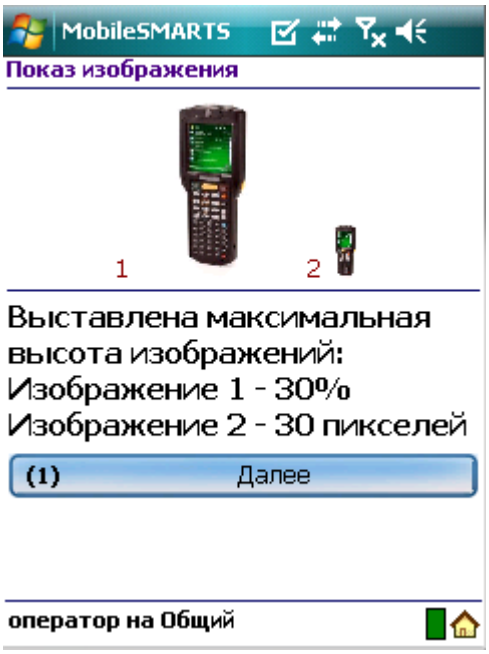
Максимальный размер (высоту и ширину) изображения можно указывать в процентах или пикселях. В данном

примере у изображения 1 указана максимальная высота в процентах, у изображения 2 указана в пикселях.

```
1 <img size="stretch" max>{картинка}</img> 2<img size="stretch" max>{картинка}</img>
```




Результат на терминале:




Если высота или ширина изображения задана в процентах, то на разных терминалах (в зависимости от размеров экрана) размер картинки будет разным.

## Прозрачность цвета фона изображения

**Пример 1 | Прозрачность цвета фона задается названием цвета**

Посмотрим, как работает прозрачность цвета фона изображения. Возьмем одно изображение. В первом случае не будем задавать прозрачность (оригинал), во втором случае добавим прозрачность зеленого цвета фона, задав атрибут «tcolor="Green"».

```
Оригинал изображения
<img size="stretch" max>{картинка2}</img><hr/>
Добавлен атрибут tcolor="Green"
Прозрачность зеленого цвета фона
<img tcolor="Green" size="stretch" max>{картинка2}</img>
```

Редактор строк

Текст:

```
Оригинал изображения
<img size="stretch" maxheight="20%">{картинка2}</img><hr/>
Добавлен атрибут tcolor="Green"
Прозрачность зеленого цвета фона
<img tcolor="Green" size="stretch" maxheight="20%">{картинка2}</img>
```

**Результат на терминале:**

**Зеленый цвет в изображении стал прозрачным.**

**Пример 2 | Прозрачность цвета фона задается кодом цвета**

**Задать прозрачности цвета фона можно не только написав цвет, но и введя код цвета #000000 (черный).**

```
Оригинал изображения
<img size="stretch" max>{картинка2}</img><hr/>
Добавлен атрибут tcolor="#000000"
Прозрачность черного цвета фона
<img tcolor="#000000" size="stretch" max>{картинка2}</img>
```

Редактор строк

Текст:

```
Оригинал изображения
<img size="stretch" maxheight="20%">{картинка2}</img><hr/>
Добавлен атрибут tcolor="#000000"
Прозрачность черного цвета фона
<img tcolor="#000000" size="stretch" maxheight="20%">{картинка2}</img>
```

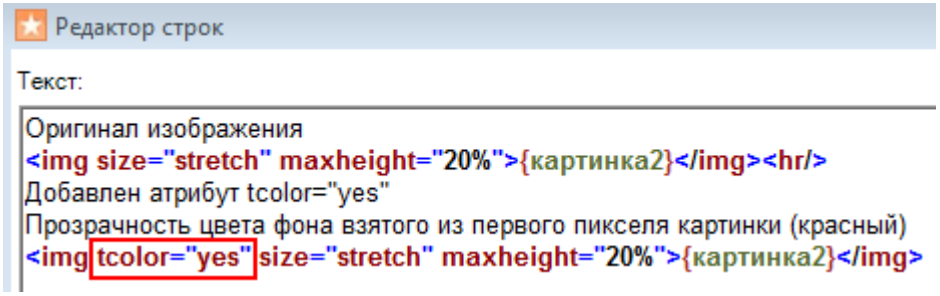
**Результат на терминале:**

**Черный цвет в изображении стал прозрачным.**

**Пример 3 | Прозрачность цвета фона задается первым пикселем изображения**

**Кроме задания конкретного цвета прозрачности фона, можно задать прозрачность, указав цвет, взятый у первого пикселя вставляемого изображения с координатами (0,0), добавив атрибут «tcolor="yes"». В данном примере это красный цвет.**

```
Оригинал изображения
<img size="stretch" max>{картинка2}</img><hr/>
Добавлен атрибут tcolor="yes"
Прозрачность цвета фона взятого из первого пикселя картинки (красный)
<img tcolor="yes" size="stretch" max>{картинка2}</img>
```




**Результат на терминале:**

Красный цвет в изображении стал прозрачным.

интерфейс, изображения, форматирование

## Не нашли что искали?

Задать вопрос в техническую поддержку

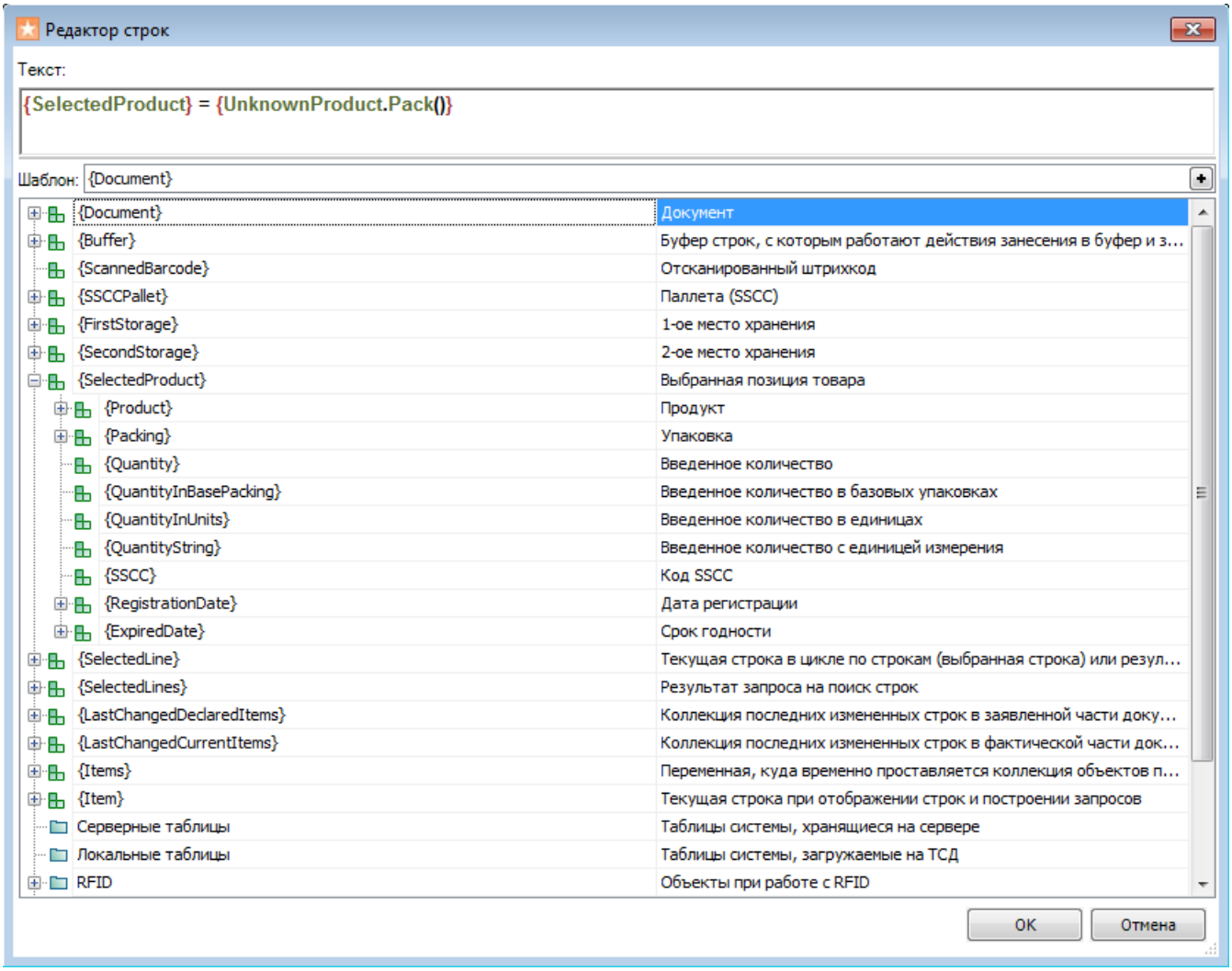
# Шаблоны текстов и выражений Mobile SMARTS

Последние изменения: 2024-03-26

Шаблон текста для отображения в окне может содержать любое количество выражений вперемешку с обычным текстом. Например, «Выбрано: {ScanedBarcode} — {SelectedProduct}». Такая строка позволит выводить на экран сообщения вида «Выбрано: 2345069545 — картофель мороженый (пачка)», где ScanedBarcode и SelectedProduct заменяются соответствующими значениями, хранящимися в сессии.

Всё до первого двоеточия ({путь к значению[:формат]}) является указанием на то, что выводить, всё после — форматом вывода значения в строку. Формат зависит от типа значения — строка это, число, дата или что-то другое. Для правильного использования формата нужно знать, каков будет тип значения. Двоеточия может и не быть, т.е. часть с форматом необязательная — если формат не указан, то используется формат по умолчанию для соответствующего типа значений.

Для написания шаблонов в редакторе текстов в «Панели управления» встроена помощь по шаблонам, куда внесены наиболее часто используемые варианты.




И синтаксис пути к значению, и синтаксис формата взяты из языка C# и спецификаций библиотеки Microsoft .NET. В .NET и, соответственно, в Mobile SMARTS всё является объектами — и строки, и числа, и более сложные вещи, такие как номенклатура, штрихкоды или складские документы. Источником данных для выражений является сессия исполнения документа, которая представляет собой просто набор переменных. Системные переменные и все переменные, заведенные разработчиком операции, складываются в одну кучу — сессию. В сессии лежит и сам обрабатываемый документ.

Поля в объекте, такие как номер документа или набор его строк, в .NET называются свойствами объекта. У

каждого такого свойства есть своё имя. Зная имя свойства можно получить значение этого свойства у конкретного объекта. Это значение также будет являться объектом, у которого есть свои свойства и т.д., при этом одно имя свойства отделяется от другого точкой. Кроме свойств у объектов есть методы — это функции, значение которых вычисляет сам объект. Синтаксис вызова метода — имя метода с круглыми скобками «()», в которых указываются передаваемые параметры. Таким образом, можно строить цепочки из обращения к свойствам и методам, чтобы получить из объекта нужные данные.

Соответственно, самый простой пример пути к значению — строка из имен свойств и методов со скобками, разделенных точками. Кроме этого выражения могут содержать арифметические операции, т.е. не просто значение свойства или вызов метода, но и сумма значений двух свойств или произведение результата вызова метода на число + sin(10).

> Следует всегда иметь в виду, что результатом обработки шаблона является текст. Поэтому, если a = 7 и b = 1, например, то не стоит путать следующие шаблоны:
> {a} - {b}» даст в результате «7 - 1», и только шаблон
> «{a - b}» даст «6».

Другой из примеров пути к значению — указание специального объекта «global::», после которого идет указание на вызов статического метода или взятие значения статического свойства (.NET), а далее можно указать путь от полученного результата.

В том случае, если система не может вычислить значения или оно равно (null), ничего выведено не будет.

При задании шаблона можно указывать шаблон вывода значения. Например, « {SelectedProduct.ExpireDate: Срок годности — (0)} » отобразит что-то вроде «Срок годности — 01 мая 2009». А «{SelectedProduct.ExpireDate: Срок годности — (0:MM$$yyyy)}» отобразит что-то вроде «Срок годности — 05$$2009».

Если срок годности не задан (т.е. если поле не существует или равно null), то не будет выведено ничего: ни даты, ни текста «Срок годности –». Это очень удобно для условного вывода значений. Можно указать целую страницу шаблонов для вывода различных возможных свойств номенклатуры, например, а на экране получить строки только для тех свойств, которые реально присутствуют. Не существующие или не проставленные свойства, а также их «окаймление» в виде разных подписей и пояснений, просто не будут отображены.

шаблоны, интерфейс

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Примеры шаблонов текстов и математических выражений Mobile SMARTS

Последние изменения: 2024-03-26

## Примеры шаблонов текстов

| Текст в шаблоне | Пример результата и описание |
|---|---|
| Накладная №{Document.Id} | |
| Накладная №1742<br><br>Выражение {Document.Id} было заменено на значение свойства с именем {Id} текущего документа, лежащего в сессии под именем «Document». | |
| Длина номера: {Document.Id.Length} цифр | |
| Длина номера: 4 цифр<br><br>Выражение {Document.Id.Length} было заменено на значение свойства «Length» объекта (текстовой строки в данном случае), полученного как значение свойства с именем «Id» у документа. | |
| Итого строк: {Document.CurrentItems.Count} | |
| Итого строк: 16<br><br>Выражение {Document.CurrentItems.Count} было заменено на значение свойства «Count» табличной части фактических строк документа. | |

| |
|---|
| {Document.CurrentItems.Count:Итого строк: (0)} |
| Итого строк: 16<br><br>То же самое. Первое двоеточие — отделяет путь от формата. Второе — просто двоеточие. «(0)» в формате было заменено на результат выражения. |
| Результат: {Document.abcdefgh ()} строк |
| Результат: строк<br><br>У документа не существует метода abcdefgh, поэтому результатом выражения будет пустая строка. |
| {Document.abcdefgh ():Результат: (0) строк} |
| У документа не существует метода abcdefgh, поэтому результатом всего выражения в скобках будет пустая строка. |
| Дата: {global:System.DateTime.Today} |
| Дата: 20.04.2009<br><br>У класса System.DateTime (дата и время) есть статическое свойство Today, которое всегда возвращает текущую дату (сегодняшнюю дату, которая сегодня). |
| {global:System.DateTime.Today.Day} ! |
| 4 !<br><br>У класса System.DateTime (дата и время) есть статическое свойство Today, которое всегда возвращает текущую дату. У текущей даты взято значение свойства Day (номер дня), а потом ко всему этому прибавились пробел и восклицательный знак. |

## Примеры шаблонов математических выражений

| Текст в шаблоне | Пример и описание |
|---|---|
| Итого: {SelectedProduct.Quantity+SelectedProduct.Quantity} шт. | |
| Итого: 15 шт.<br><br>Выражение «SelectedProduct.Quantity» было заменено на значение свойства «Quantity» (количество) и выполнено сложение. | |
| Итого: {SelectedProduct.Quantity:(0)}+{SelectedProduct.Quantity:(0)} шт. | |
| Итого: 9+6 шт.<br><br>В данном примере «+» (знак плюс) написан как текст и ничего не суммирует. | |
| {Document.DeclaredItems.DeclaredQuantity- Document.DeclaredItems.CurrentQuantity: Осталось набрать: (0) шт.} | |
| Осталось набрать: 6 шт.<br><br>Получили разницу планового количества и фактического. | |
| {SelectedProduct.ЦенаСклад: Цена: (0) с}<br><br>{SelectedProduct.Quantity: Количество: (0) шт.}<br><br>Сумма: {SelectedProduct.ЦенаСклад*SelectedProduct.Quantity: (0) с} | |
| Цена: 1035 р.<br><br>Количество: 5 шт.<br><br>Сумма: 5175 р.<br><br>Вывели цену и количество товара и, перемножив их значения «(0)» получили сумму.<br><br>Первое двоеточие — отделяет путь от формата. Второе — просто двоеточие. «с» — означает р. (рубли). | |

| |
|---|
| {SelectedProduct.ЦенаСклад: Цена: (0) с}<br><br>{SelectedProduct.Quantity: Количество: (0) шт.}<br><br>Сумма: {SelectedProduct.ЦенаСклад*SelectedProduct.Quantity: (0) с} |
| Если у товара не указана цена, то результат будет таким:<br><br>Количество: 5 шт.<br><br>На экран будет выведено только количество товара. |

**Если при разборе строки из имен свойств и методов будет обнаружено, что таких свойств или методов нет или нет объектов, у которых их следует брать, в качестве результата выражения ничего не будет отображено.**

**По ссылкам ниже приведены описание доступных методов по работе с числами, строками и датами в .NET (на русском):**

http://msdn.microsoft.com/ru-ru/library/system.string.aspx

http://msdn.microsoft.com/ru-ru/library/system.int32.aspx

http://msdn.microsoft.com/ru-ru/library/system.double.aspx

http://msdn.microsoft.com/ru-ru/library/system.decimal.aspx

http://msdn.microsoft.com/ru-ru/library/system.datetime.aspx

шаблоны, выражения

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Шаблоны Mobile SMARTS с условием

Последние изменения: 2024-03-26

Использование условий в шаблонах серьёзно упрощает жизнь разработчику на Mobile SMARTS. Условия в шаблонах позволяют избежать лишнего применения действия «Проверка условий».

Рассмотрим синтаксис условий и примеры их использования.

```
{Условие:результат_истина;результат_ложь}
```

**Условие** – некоторое булево выражение или переменная, принимающая значения true или false. Условие может состоять из нескольких условий, в таком случае они объединяются с помощью логических операторов:

- || - или;
- && - и.

**результат_истина** – результат работы шаблона когда выражение условия принимает значение true;

**результат_ложь** – результат работы шаблона когда выражение условия принимает значение false.

Значения результатов могут быть также заданы шаблонами с условием.

Итогом использования этого шаблона будет вывод результата, соответствующего значению условия. Если условие принимает значение null, то ни один результат выведен не будет.

Рассмотрим примеры.

## Пример составного условия

Шаблон используется в конфигурации «Mobile SMARTS: Магазин 15» для показа кнопки выбора склада в зависимости от нескольких условий.

Значение шаблона: {Document.CreatedOnPDA || Document.СкладВведенНаТСД || Document.КодСклада==null: Склад: {Document.СкладОтобр};}

Шаблон размещён в тексте кнопки «Меню».

Разберёмся, как это работает . Для выражения условия в примере используется несколько переменных:

**Document.CreatedOnPDA** – true, если документ создан на мобильном устройстве, если нет – false;

**Document.СкладВведенНаТСД** – true, если склад был выбран на мобильном устройстве, если нет – false;

**Document.КодСклада==null** – выражение сравнения поля документа КодСклада с null. True, если КодСклада – null.

Все составляющие общего условия объединены между собой логическим оператором «или» («||»). Это значит, что итоговое значение всегда будет true, кроме случая, когда все три элемента будут иметь значение false.

Иными словами, кнопка будет выводиться всегда, кроме случая, когда документ выгружен из учётной системы, имеет заполненное поле шапки КодСклада и склад не выбирался на мобильном устройстве. В этом случае кнопка отображаться не будет.

## Вывод текста в зависимости от условия

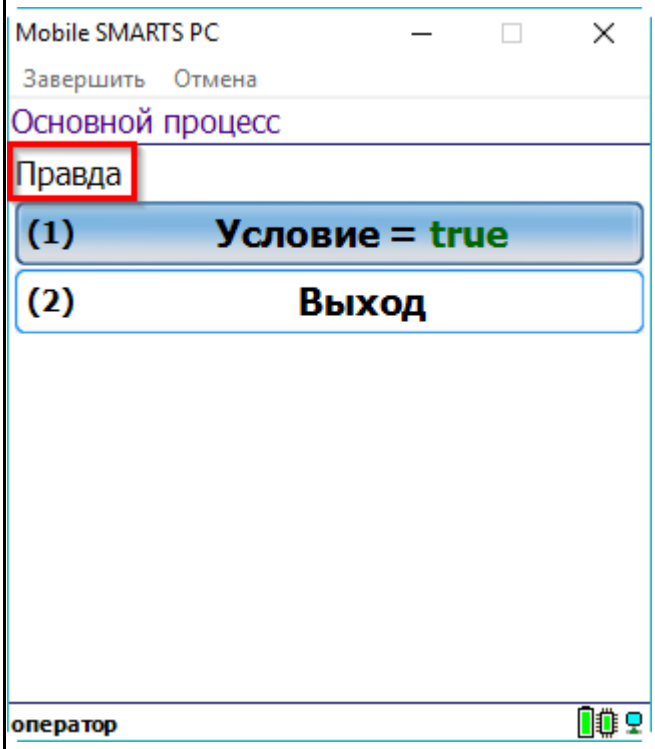
Текст в самом окне: {Условие:Правда;Ложь}

Текст на первой кнопке: {Условие:true;false}

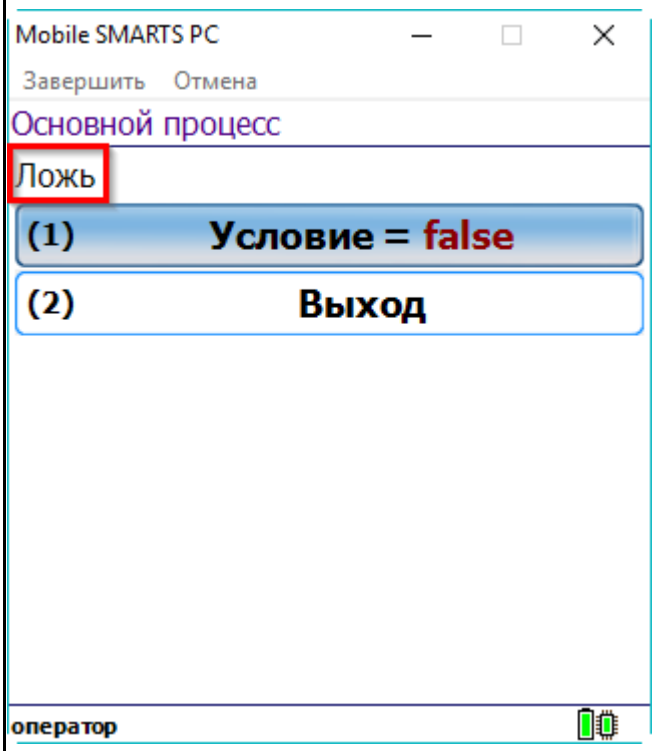
Текст на второй кнопке: «Выход»

Результат работы:

Условие = true




Условие = false



## Скрытие текста по значению условия

Текст в самом окне: {Условие:Значение условия - {Условие};}

Текст на первой кнопке: Условие = {Условие:true;false}

Текст на второй кнопке: «Выход»

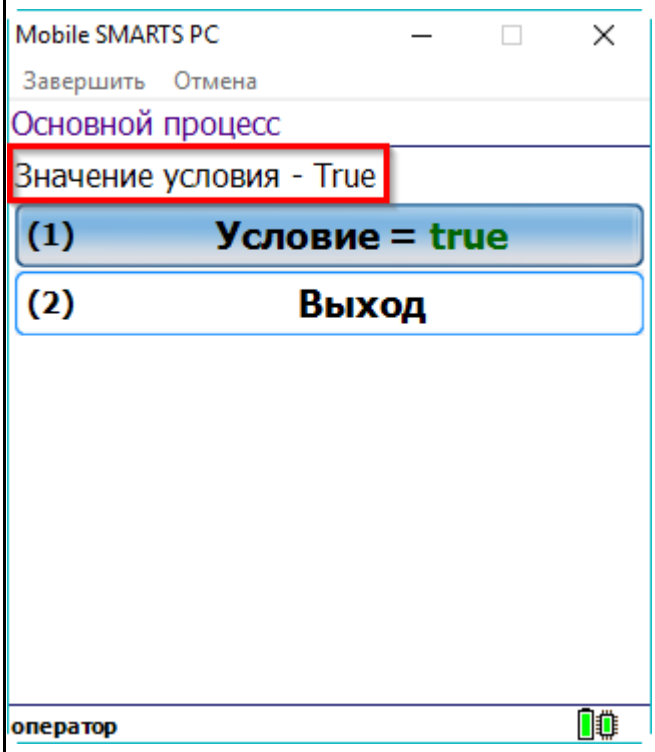
Обратите внимание, что результат, который будет выведен при значении условия true, может содержать шаблоны для вывода значений переменных, в данном случае это переменная Условие.

Результат, выводимый при значении false, отсутствует. Это позволяет ничего не выводить на экран, когда условие принимает значение false.

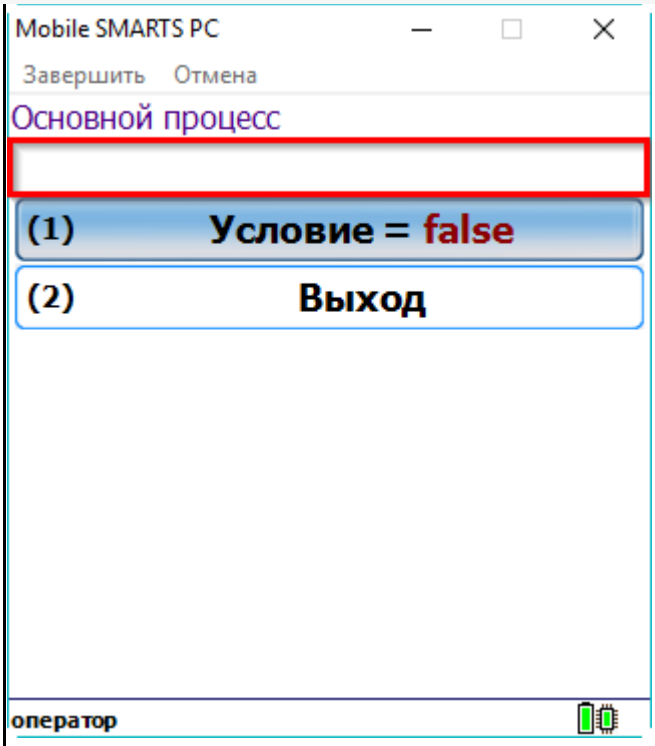
Шаблон размещён в свойстве «Текст» в самом окне действия «Меню».

Результат работы:

Условие = true



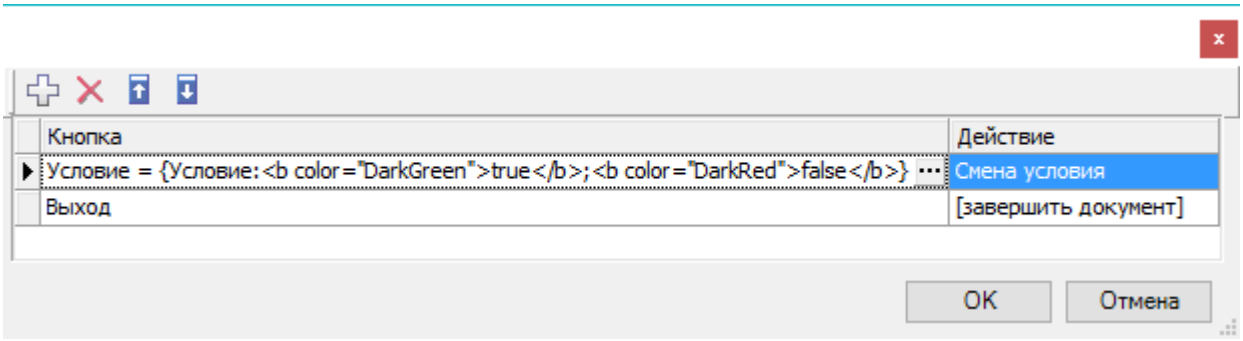

Условие = false



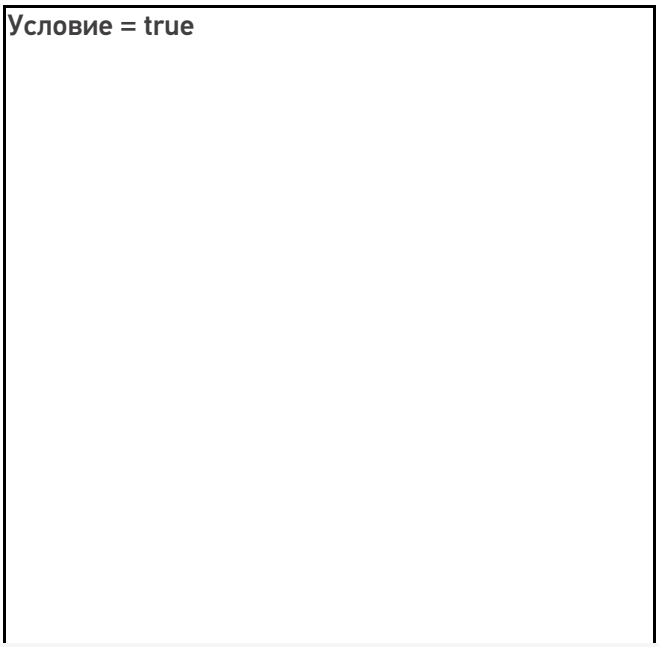

## Вывод форматированного текста на кнопках меню

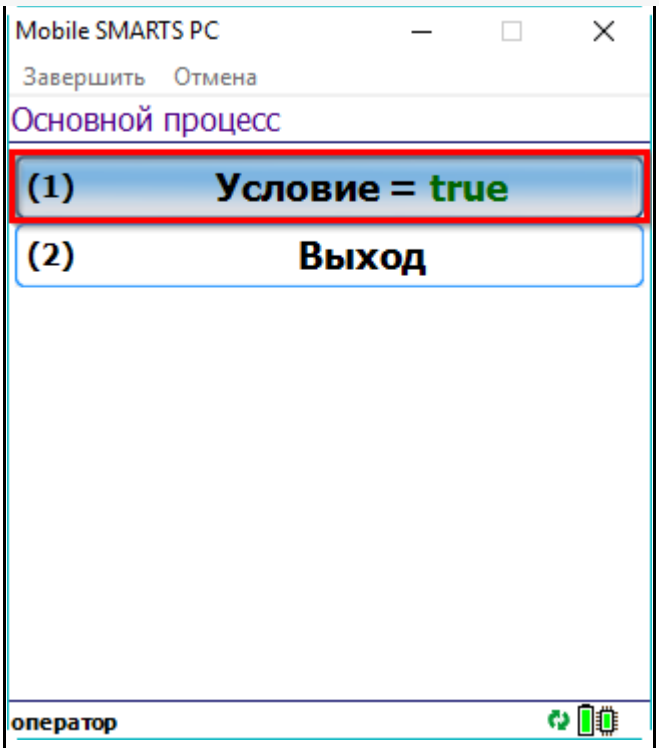
Такой вариант использования отлично подходит, например, для вывода текущих настроек на кнопке.

Текст на кнопке: Условие = {Условие:true;false}

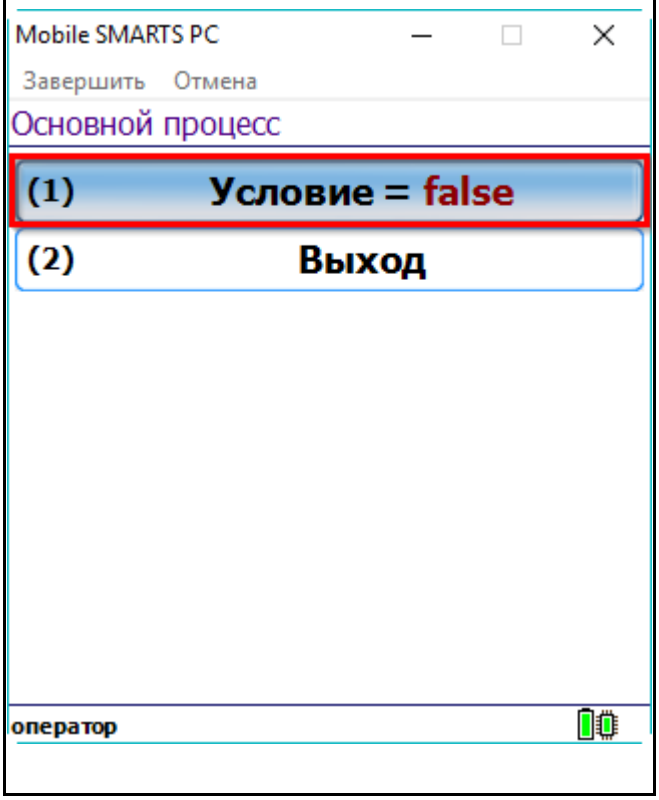



Результат работы:






Условие = false



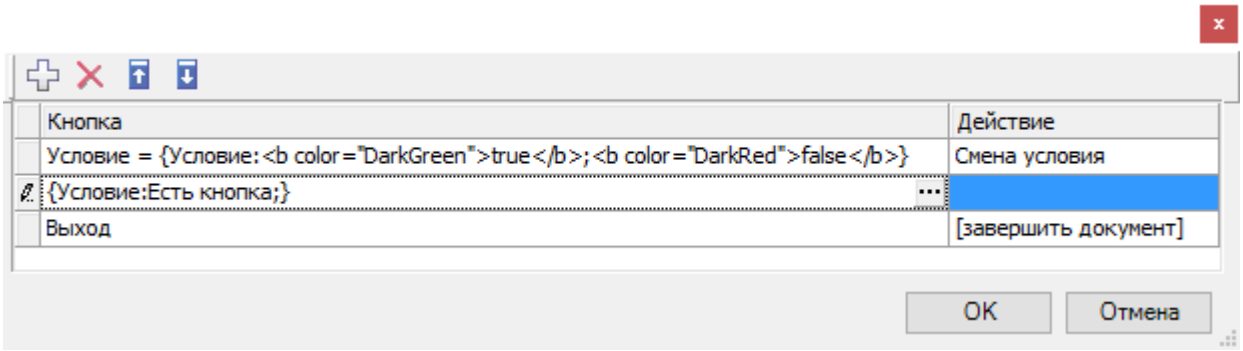

## Показ и скрытие кнопок меню в зависимости от условия

Такой вариант использования подходит например для вывода кнопки ввода склада, при условии, что склад в документе не был проставлен ранее.

Значение шаблона: {Условие:Есть кнопка;}

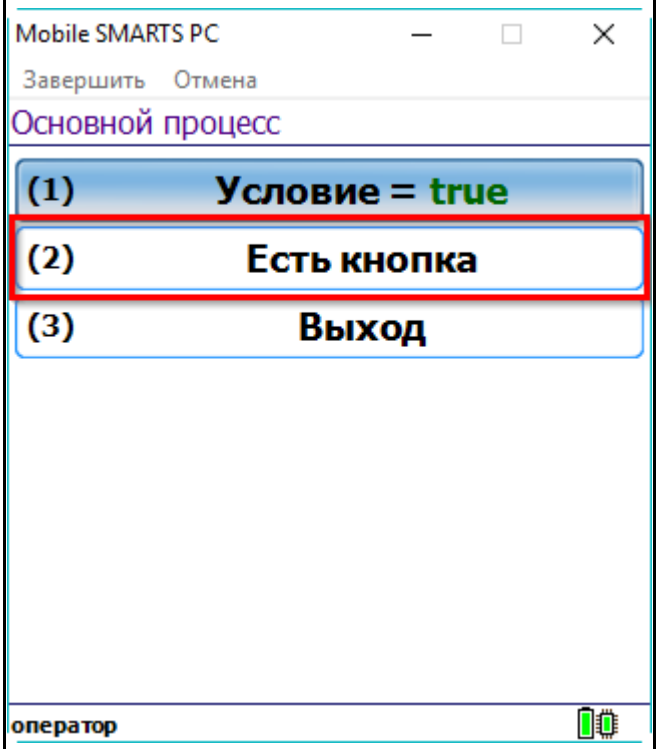
Как видите, при значении условия false, никакой текст выводиться не должен. Одновременно с текстом исчезает и появляется кнопка.

Шаблон размещён в тексте кнопки «Меню»:

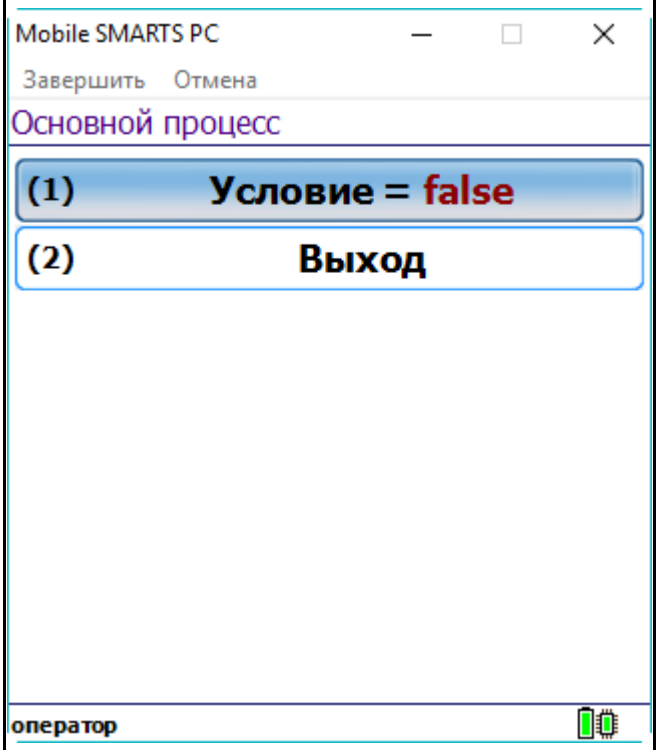



**Результат работы:**

Условие = true




Условие = false




шаблоны

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Основы верстки в Mobile SMARTS

Последние изменения: 2024-03-26

Методология разработки в Mobile SMARTS такова, что функционал приложения изначально неотделим от пользовательского интерфейса.

Чтобы сформировать внешний вид приложения (текст, картинки, поля ввода данных), который впоследствии будет отображаться на экране мобильного устройства, мы используем элементы HTML. Они позволяют задать стиль неопределенных в дереве элементов документа, а также генерировать содержимое, которого нет в исходном коде текста.

К тегам форматирования текста в HTML можно отнести теги, изменяющие отображение выделенного фрагмента. Использование форматирования позволяет отобразить информацию на экране в удобном для клиента виде.

**Основная задача верстки** — сделать интерфейс приложения максимально простым и понятным пользователю, поэтому мы предоставили возможность менять внешний вид приложений на платформе Mobile SMARTS «под клиента» с помощью следующих инструментов:

1. **HTML-теги** — используются для форматирования текста. С их помощью можно задать необходимый цвет, размер, стиль, что позволяет отобразить информацию на экране в удобном пользователям виде, выделить важные сообщения в тексте и т. п.

   Теги строятся по принципу: <имя тега>. Имя тега может состоять из английских букв и цифр. Теги обычно пишутся парами — открывающий тег и соответствующий ему закрывающий. Разница между открывающим и закрывающим тегами в том, что в закрывающем теге после открывающей угловой скобки стоит слеш.

2. **Классы** — позволяют менять стиль элемента в зависимости от действий пользователя, например, поменять цвет кнопки, по которой уже осуществлялся переход.

3. **Атрибуты тегов и события.** Позволяют задать значения для свойств, применимые к данному тегу. Атрибуты размещаются внутри открывающего тега.

   Значения HTML-атрибутов всегда пишутся в двойных кавычках.

На платформе Mobile SMARTS применяются две группы верстки:

- упрощенная верстка;
- полнофункциональная верстка.

> С момента выхода 3.3 платформы у Android-клиента появились встроенные отступы от краев экрана. Данные отступы отключить нельзя.

Не нашли что искали?

Задать вопрос в техническую поддержку

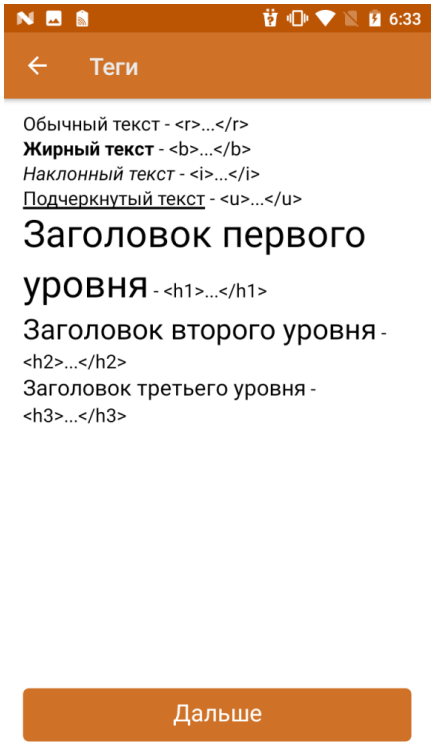
# Упрощенная верстка в Mobile SMARTS

Последние изменения: 2024-03-26

Для простого форматирования текста можно использовать теги упрощённой верстки. Основными тегами упрощенной верстки являются:

- <r> — обычный шрифт;
-  — жирный шрифт;
- <u> — подчеркнутый шрифт;
-  — наклонный шрифт;
- <h1> — заголовок первого уровня;
- <h2> — заголовок второго уровня;
- <h3> — заголовок третьего уровня.

Для вышеописанных тегов необходимо использовать закрывающий тег, например:




<r>Обычный текст</r>

Жирный текст

Наклонный текст

<u>Подчеркнутый текст</u>

<h1>Заголовок первого уровня</h1>

<h2>Заголовок второго уровня</h2>

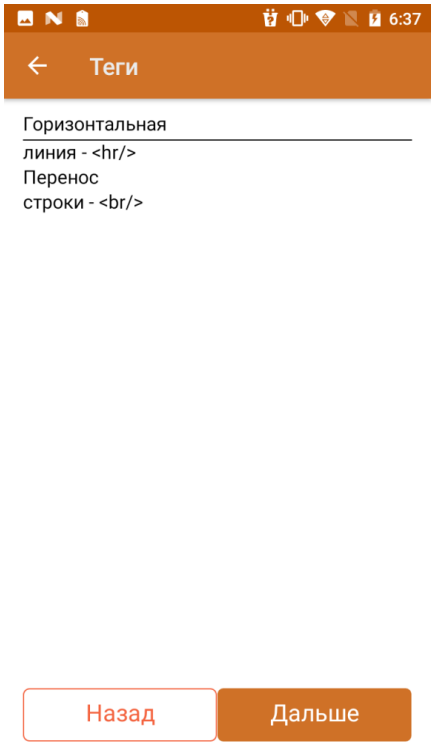
<h3>Заголовок третьего уровня</h3>

Также, существуют теги упрощенной верстки, для которых закрывающий тег не нужен:

- <br/> — перенос строки;

- <hr/> — горизонтальная линия

Пример простого синтаксиса:

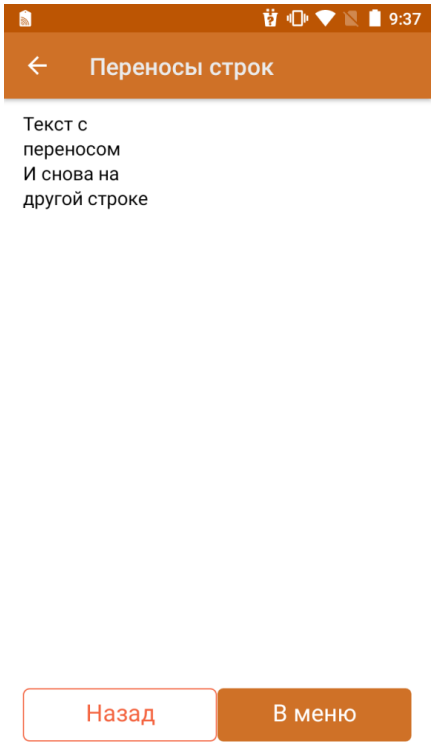



Горизонтальная<hr/>линия — \<hr/>

Перенос<br/>строки — \<br/>

Основное отличие упрощенной верстки заключается в правилах формирования отображения: при полнофункциональной верстке переносы строк игнорируются, в то время как в упрощенной нет.

Пример простого синтаксиса:

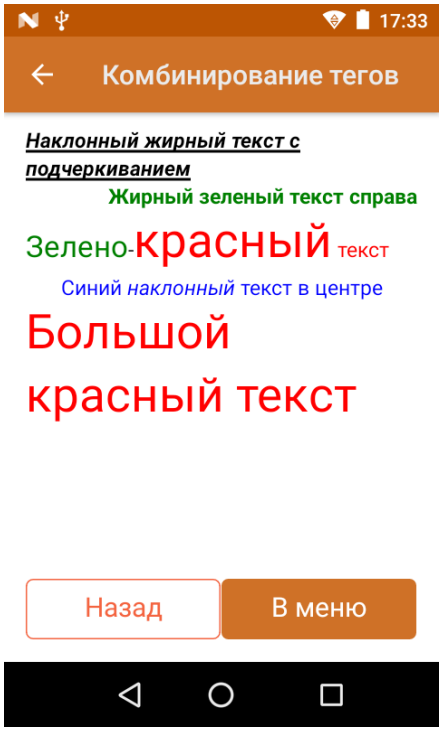



Текст с <br/> переносом
И снова на
другой строке

В случае, если теги упрощенной верстки используются внутри тега , данная верстка будет являться и обрабатываться по правилам блочной верстки.

Для формирования правильного отображения теги можно комбинировать между собой. Главное, следить за тем, где находится закрывающий тег. В следующем примере теги используются совместно с атрибутами( подробнее в статье «Использование атрибутов тегов»).

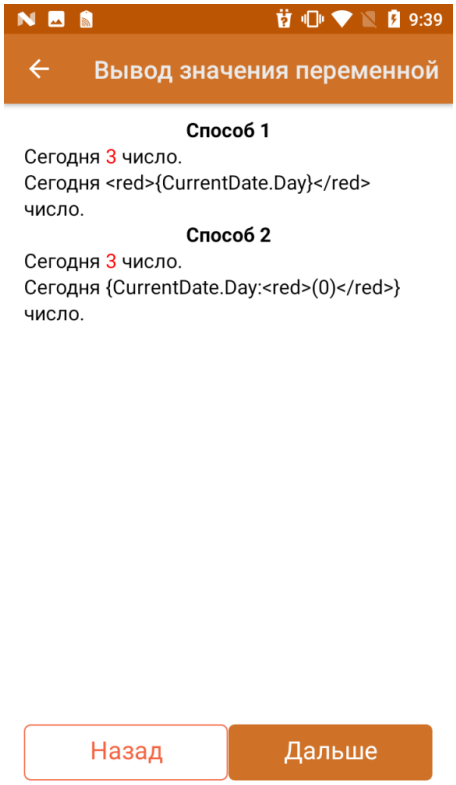
Пример простого синтаксиса:




```
<b><u><i>Наклонный жирный текст с подчеркиванием</i></u></b>

<b color="green» align="right">Жирный зеленый текст справа</b>

<green size="10">Зелено</green>-<red size="14">красный</red> <red>текст</red>

<r color="#0000FF» align="center">Синий <i>наклонный</i> текст в центре</r>

<red size="14">Большой красный текст</red>
```

Вышеуказанные теги можно применять при отображении данных из переменной/свойств объектов.

Пример простого синтаксиса:




**Способ 1**

**Сегодня <red>{CurrentDate.Day}</red> число.**

**Сегодня \<red>\{CurrentDate.Day\}\</red> число.**

**Способ 2**

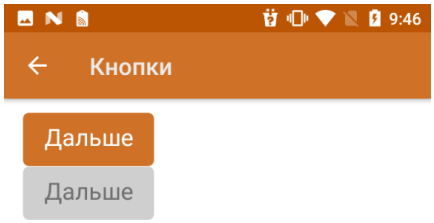
**Сегодня {CurrentDate.Day:<red>(0)</red>} число.**

**Сегодня \{CurrentDate.Day:\<red>(0)\</red>\} число.**

**В отличие от вышеуказанных тегов, которые используются для формирования отображения текста в упрощенной верстке, также могут использоваться следующие теги: <button> и <img>.**

**Тег <button> используется для создания кнопки. Обязательно должен иметь завершающий тег. С помощью данного тега можно создать дополнительные кнопки управления в различных частях окон.**

**Примеры использования данного тега:**




```
<button direction="">Дальше</button>

<button direction=»» enabled="false">Дальше</button>

<button direction=»» visible="false">Дальше</button>
```

С помощью атрибута direction указывается, на какое действие алгоритма будет совершен переход при нажатии. Атрибуты enabled и visible управляют доступностью нажатия/отображения элемента.

Тег <img> используется для отображения изображения. Обязательно должен иметь завершающий тег.
С помощью данного тега можно отображать изображения из файла ресурсов, по ссылке, через переменную из таблицы/по ссылке.

Пример простого синтаксиса:

{logo = "https://www.cleverence.ru/local/templates/cleverence/img/logo.png"}

<img>https://www.cleverence.ru/local/templates/cleverence/img/logo.png</img>

<img>logo.png</img>

<img>testres.logo</img>

<img>{logo}</img>

Само изображение для отображения может быть задано несколькими путями:

| |
|---|
| **Шаблон** |
| **Путь к файлу** |
| **Описание** |

| |
|---|
| <img>\Images\picture.jpg</img><br><br><img>\Flash\Images\picture.jpg</img> |
| \Images\picture.jpg<br><br>\Flash\Images\picture.jpg |
| Изображение находится по заданному абсолютному пути на терминале |
| <img>picture.jpg</img><br><br><img>Images\picture.jpg</img> |
| \Application\MobileSMARTS\picture.jpg<br><br>\ Application\MobilesSMARTS\Images\<br><br>picture.jpg |
| Изображение ищется по пути<br><br><Папка программы на терминале>\<заданный относительный путь><br><br>если не найдено ищется по пути<br><br><Папка программы на терминале>\<папка базы на терминале>\<заданный относительный путь> |
| <img>{ПеременнаяСПутем}</img> |
| \Flash\Images\pic1.jpg<br><br>Переменная в сессии {ПеременнаяСПутем}=”\Flash<br><br>\Images\pic1.jpg” |
| Изображение ищется по пути, лежащему в переменной сессии. |

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Полнофункциональная верстка в Mobile SMARTS

Последние изменения: 2024-03-26

Признаком использования полнофункциональной верстки является использование тега . С помощью этого тега можно настроить визуальное расположение вкладываемых в него элементов, а также размер элемента/цвет фона и другие.

Для одного свойства визуального действия доступен только один внешний . При использовании тега  всё визуальное отображение в свойстве визуального действия, которое написано вне тега, отображено не будет, однако, вычисляемые выражения будут выполнены.

Подробное описание тега с примерами смотрите в статье «Тег ».

## Тег <table>

Данный тег служит контейнером для элементов, определяющих содержимое таблицы и для размещения 2х блоков на одной плоскости. Любая таблица состоит из строк, колонок и ячеек, которые задаются с помощью тегов <tr> и <td>.

Тег <tr> используется для разметки строки таблицы.

Тег <td> предназначен для создания одной ячейки таблицы. Данный тег должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

> Все перечисленные теги должны обязательно иметь завершающий тег.

Подробное описание тегов с примерами смотрите в статье «Теги <table>, <tr> и <td>».

## Тег <input>

Также существует возможность добавить поле ввода прямо в верстке.

Для этого нужно воспользоваться тегом <input>. У данного тега есть основной атрибут type, с помощью которого определяется, как будет выглядеть поле ввода.

Подробное описание тега с примерами смотрите в статье «Тег <input>».

## Использование атрибутов тегов

HTML-теги могут содержать один или несколько атрибутов. Атрибуты добавляются в тег для того, чтобы информировать браузер о том, как данный тег должен отображаться в приложении. Атрибуты задаются в начальном теге элемента и состоят из имени и значения, которые отделяются друг от друга знаком равно (=).

В HTML используются «двойные кавычки», например синтаксис применения атрибута к тегу:

```
<tag attribute="value"></tag>
```

Некоторые атрибуты стандартны для большинства html-тегов, это так называемые общие атрибуты, а есть частные атрибуты, которые используются только для определенных тегов/

Наиболее часто используемыми атрибутами являются универсальные атрибуты (например class, id), определяющие общие свойства элементов, и локализующие атрибуты, которые указывают на свойства языка написания содержимого элемента.

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Общие атрибуты тегов в Mobile SMARTS

Последние изменения: 2024-03-26

**Общие (глобальные, универсальные) атрибуты** применяются практически ко всем элементам HTML, поэтому выделены в отдельную группу, чтобы не повторять их для каждого элемента.

Список атрибутов;

- align
- width/height
- class
- id
- color
- maxlines
- style

| Атрибут |
| --- |
| Значение |
| Описание |
| align |
| right, left, center |
| горизонтальное выравнивание |

Для атрибута align синтаксис будет выглядеть следующим образом:

```
<b align="center">Атрибут "align"</b>

<r align="left">Текст слева</r>

<r align="left">align="left"</r>

<r align="center">Текст в центре</r>

<r align="center">align="center"</r>

<r align="right">Текст справа</r>

<r align="right">align="right"</r>
```

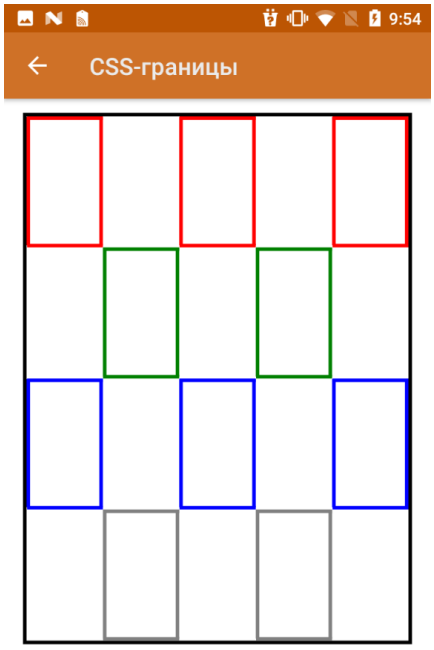
| Атрибут |
| --- |
| Значение |
| Описание |
| width |
| числовое значение |
| определяет ширину таблиц, изображений или ячеек таблицы |
| height |
| числовое значение |
| определяет высоту таблиц, изображений или ячеек таблицы |

На примере форматирования кнопки для атрибутов width и height синтаксис будет выглядеть следующим образом:

```
<button direction="4" width="100%" height="15%">Дальше</button>
```

| Атрибут |
| --- |
| Значение |
| Описание |
| class |
| правило класса<br>или стиль класса |
| Задает стилевой класс, который позволяет связать определенный тег со стилевым оформлением.<br>В значении допускается указывать сразу несколько классов, разделяя их между собой пробелом |

Для атрибута class синтаксис будет выглядеть следующим образом:




CSS: class="имя_класса"

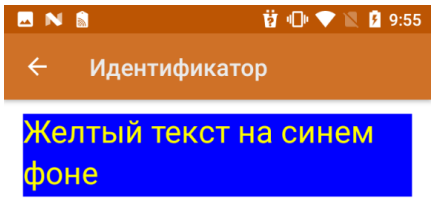
Еще

```
CSS: class="имя_класса"
<div width="100%">
<table width="100%" style="border:3dp solid black;" >
<tr>
<td width="20%" height="20%" class="redBorder"></td>
<td></td>
<td class="redBorder"></td>
<td></td>
<td class="redBorder"></td>
</tr>
<tr>
<td width="20%" height="20%"></td>
<td class="greenBorder"></td >
<td ></td>
<td class="greenBorder"></td>
<td></td>
</tr>
<tr>
<td width="20%" height="20%" class="blueBorder"></td>
<td></td>
<td class="blueBorder"></td>
<td></td>
<td class="blueBorder"></td>
</tr>
<tr>
<td width="20%" height="20%"></td>
<td class="grayBorder"></td>
<td></td>
<td class="grayBorder"></td>
<td></td>
</tr>
</table>
</div>
```

| Атрибут |
| --- |
| Значение |
| Описание |

| id |
|---|
| идентификатор должен обязательно начинаться с латинского символа и может содержать в себе латинские буквы (A-Z, a-z), цифры (0-9), символ дефиса (-) и подчеркивания (_). Использование русских букв в именах идентификатора недопустимо. |
| задает стилевой идентификатор — уникальное имя элемента, которое используется для изменения его стиля; идентификатор в коде документа должен быть в единственном экземпляре, иными словами, встречаться только один раз. |

Для атрибута id синтаксис будет выглядеть следующим образом:




```
CSS:
#example{
font-size: 12pt;
background-color: blue;
color: yellow;
}
```

```
<div id="example">Желтый текст на синем фоне</div>

<div id="example">Здесь тоже используется этот идентификатор, но этот блок не отобразится</div>
```

| Атрибут |
|---|
| Значение |
| Описание |
| color |
| цвет |
| устанавливает цвет текста, используя либо название цвета, либо шестнадцатеричный формат #RRGGBB (см. Таблица цветов в HTML) |

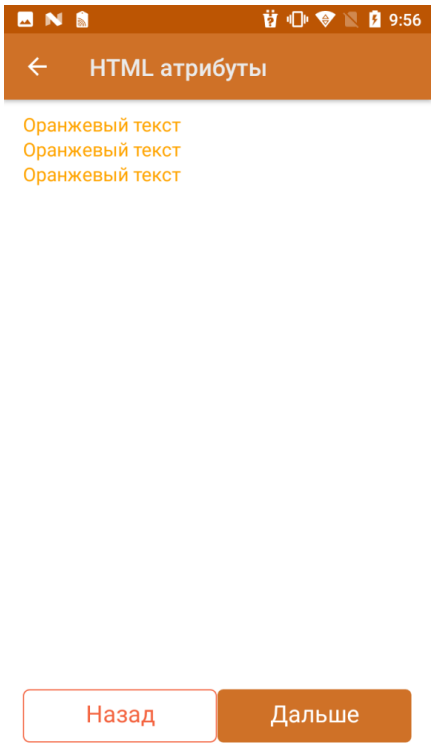
Для задания цвета текста используется тег с именем цвета или необязательный атрибут color="...".

> Если атрибут не задан, то для вывода текста используется цвет по умолчанию, в соответствии со стилем отображения.

Цвет может задаваться тремя способами:

- тег — имя цвета;
- название цвета на английском языке;
- код цвета в шестнадцатеричном виде.

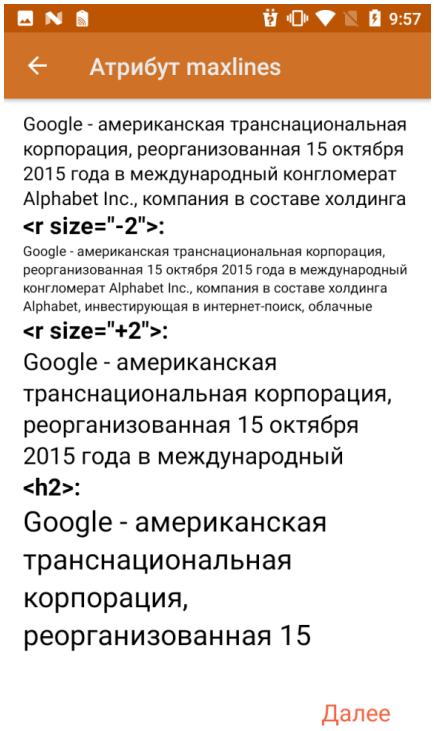
Для атрибута color в зависимости от варианта введения метаданных синтаксис будет выглядеть следующим образом:




```
<Orange>Оранжевый текст</Orange>
<r color="Orange">Оранжевый текст</r>
<r color="#FFA500">Оранжевый текст</r>
```

| Атрибут |
| --- |
| Значение |
| Описание |
| maxlines |
| числовое значение |
| обрезает текст до N строк |

Для атрибута maxlines синтаксис будет выглядеть следующим образом:




```
<div align="left">
<r maxlines="4">Google — американская транснациональная корпорация, реорганизованная 15 октября 2015
года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая
в интернет-поиск, облачные вычисления и рекламные технологии.</r><br />
<b size="+2">\<r size="-2">:</b><br />
<r size="-2" maxlines="4">Google — американская транснациональная корпорация, реорганизованная
15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet,
инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии.</r><br />
<b size="+2">\<r size="+2">:</b><br />
<r size="+2" maxlines="4">Google — американская транснациональная корпорация, реорганизованная
15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet,
инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии.</r><br />
<b size="+2">\<h2>:</b><br />
<h2 maxlines="4">Google — американская транснациональная корпорация, реорганизованная 15 октября 2015
года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая
в интернет-поиск, облачные вычисления и рекламные технологии.</h>
</div>
```

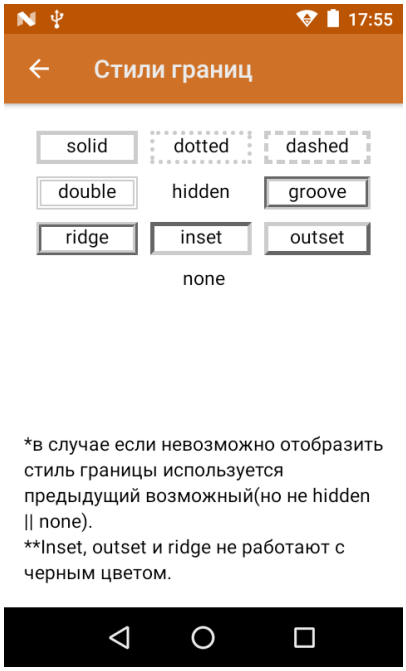
Ограничить длину выводимого текста можно также с помощью форматирования:

{SomeString = "Google — американская транснациональная корпорация, реорганизованная 15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии."}

{SomeString:Исходная строка: (0)}
<br /><br />
{SomeString:Обрезана до 100 символов: (0:T100)}
<br /><br />
{SomeString:Обрезана до 100 символов + троеточие: (0:E100)}
<br /><br />
{SomeString:Начало и конец + троеточие в середине: (0:M100)}

| Атрибут |
| --- |
| Значение |
| Описание |
| style |
| в качестве значений указываются стилевые правила |
| применяется для определения стилей элементов с помощью правил CSS |

Для атрибута style синтаксис будет выглядеть следующим образом:



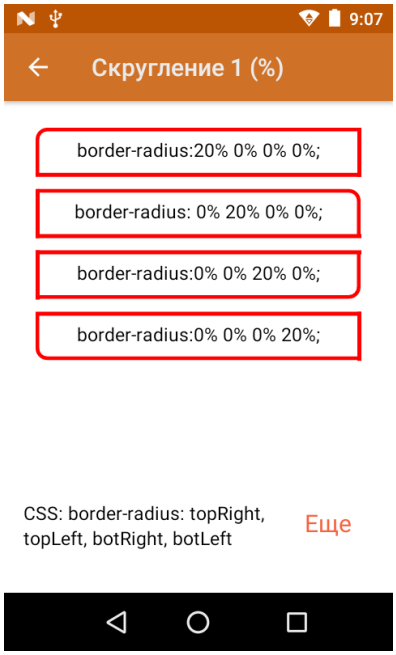

```
<div>
<table cols="3" width="100%" cellspacing="10dp" align="center" valign="middle">
<tr>
<td style="border:3dp solid #cccccc;" >solid</td>
<td style="border:3dp dotted #cccccc;">dotted</td>
<td style="border:3dp dashed #cccccc;">dashed</td>
</tr>
<tr>
<td style="border:3dp double #cccccc;">double</td>
<td style="border:3dp hidden #cccccc;">hidden</td>
<td style="border:3dp groove #cccccc;">groove</td>
</tr>
<tr>
<td style="border:3dp ridge #cccccc;">ridge</td>
<td style="border:3dp inset #cccccc;">inset</td>
<td style="border:3dp outset #cccccc;">outset</td>
</tr>
<tr>
<td colspan="3" style="border:3dp none #cccccc;">none</td>
</tr>
</table>
</div>
```

С помощью стилевого правила border-radius атрибута style можно устанавливать радиус скругления углов рамок. Можно использовать одно, два, или четыре значения.

В зависимости от количества значений скругление будет применяться (по очереди):

- 1 — для всех четырех углов;
- 2 — первое значение определяет радиус скругления верхнего левого и нижнего правого, второе — верхнего правого и нижнего левого углов;
- 3 — первое значение определяет радиус скругления верхнего левого угла, второе — одновременно для верхнего правого и нижнего левого, третье — для нижнего правого
- 4 — верхнего левого, верхнего правого, нижнего правого, нижнего левого.

В качестве значения принимаются числа в поддерживаемом формате (рекомендуется использовать dp) или проценты. Синтаксис выглядит следующим образом:




```
<div>
<table width="100%" align="center" style="vertical-align: middle;" cellspacing="10dp">
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:20% 0% 0% 0%; ">
border-radius:20% 0% 0% 0%; </td>
</tr>
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:0% 20% 0% 0%; ">
border-radius: 0% 20% 0% 0%; </td>
</tr>
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:0% 0% 20% 0%; ">
border-radius:0% 0% 20% 0%; </td>
</tr>
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:0% 0% 0% 20%; ">
border-radius:0% 0% 0% 20%; </td>
</tr>
</table>
</div>
```

Еще один пример с использованием различного количества значений:

```
<div>
   <table width="100%" align="center" valign="middle" cellspacing="10dp">
   <tr>
   <td>
   CSS: border-radius: topRight+topLeft, botRight + botLeft
   </td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp dotted red; border-radius:15dp 0dp; ">
   border-radius:15dp 0dp;
   </td>
   </tr>
   <tr>
   <td>
   CSS: border-radius: all
   </td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp inset red; border-radius:15dp; ">
   border-radius: 15dp;
   </td>
   </tr>
   <tr>
   <td>CSS: border-radius: topLeft, topRight, botRight, botLeft</td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp outset red; border-radius:5dp 10dp 15dp
20dp; ">border-radius: 5dp 10dp 15dp 20dp;</td>
   </tr>
   <tr>
   <td>CSS: border-radius: topLeft, topRight + botLeft, botRight</td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp inset red; border-radius:5dp 10dp 20dp;
">border-radius: 5dp 10dp 20dp;</td>
   </tr>
   </table>
   </div>
```

форматирование

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Частные атрибуты тегов в Mobile SMARTS

Последние изменения: 2024-03-26

Многие атрибуты в HTML являются общими для всех элементов, однако большинство из них являются специфическими для данного элемента или группы элементов. Это так называемые **частные атрибуты**.

Атрибуты обеспечивают дополнительную информацию об элементе, при этом они всегда определяются в начальном теге независимо от того парный это тег, либо одиночный. Пользователь не может создавать свои собственные атрибуты или использовать значения, не определенные спецификацией, так как это может вызывать проблемы правильной интерпретации.

Рассмотрим частные атрибуты подробнее с примерами.

- Тег 
- Тег <p>
- Теги <table>, <tr>, <td>
- Тег <input>
- Тег <img>
- Тег <button>
- Тег <a>

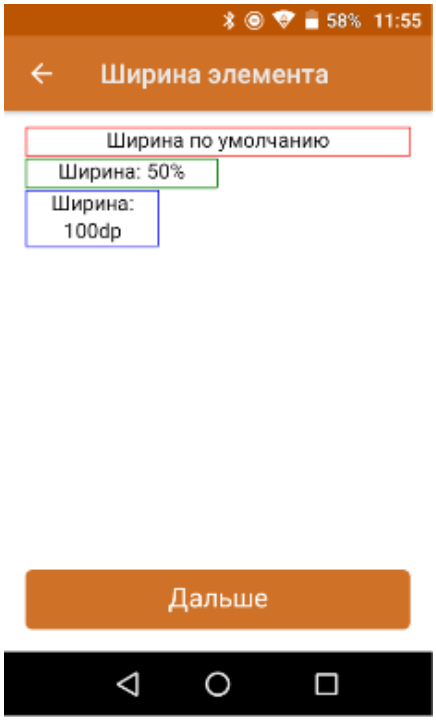
## Тег  и его частные атрибуты

Пример простого синтаксиса:

```
Текст вне тега div
{text = "Текст из переменной"}
<div>
<p>{text}</p>
<p>Текст внутри тега div</p>
</div>
```

По умолчанию, при использовании тега  используется вся доступная область экрана, изменить размер можно с помощью атрибута width.

Пример простого синтаксиса:




```
<div>
<div style="border:1dp solid red;">
Ширина по-умолчанию
</div>
<div style="border:1dp solid green; width:50%;">
Ширина: 50%
</div>
<div style="border:1dp solid blue; width:100dp;">
Ширина: 100dp;
</div>
</div>
```

Как видим из примера, блоки  размещаются вертикально.

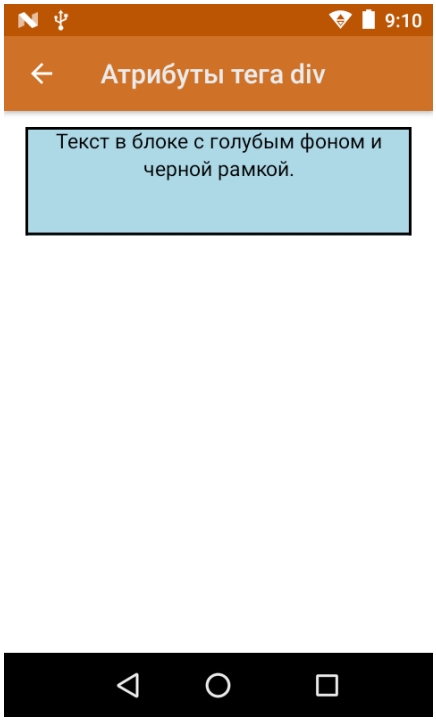
Для тега  доступны следующие частные атрибуты:

| Атрибут | Значение | Описание |
|---|---|---|
| align | right, left, center | горизонтальное выравнивание |
| bgcolor | цвет | определяет цвет фона блока/контейнера |
| border | числовое значение | определяет толщину границ блока/контейнера |

| |
|---|
| height |
| числовое значение |
| определяет высоту |

| |
|---|
| maxlines |
| числовое значение |
| обрезает текст до N строк |

Еще пример использования атрибутов тега :

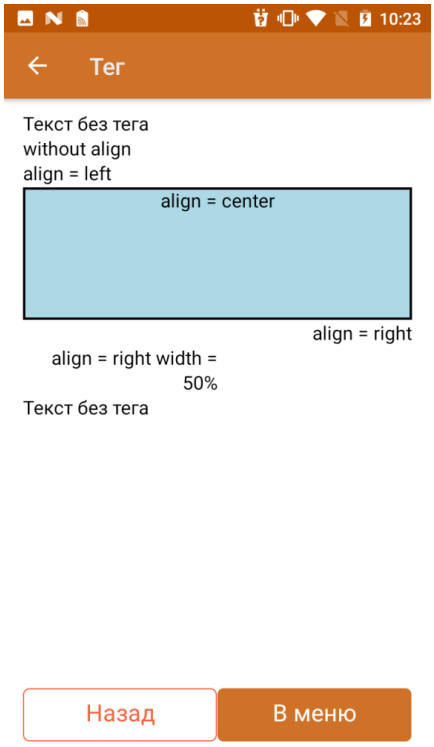



```
<div height="20%" align="center" bgcolor="lightblue" border="2dp">Текст в блоке с голубым фоном и черной рамкой.</div>
```

## Тег <p> и его частные атрибуты

Представляет собой абзац. По умолчанию использует всю доступную область родительского элемента.

Пример синтаксиса:




```
<div>

Текст без тега<p>without align</p><p align="left">align = left</p><p align="center" height="20%"
bgcolor="lightblue" border="2dp">align = center</p><p align="right">align = right</p><p width="50%"
align="right">align = right width = 50%</p>Текст без тега

</div>
```

Для тега <p> доступны следующие частные атрибуты:

| Атрибут | Значение | Описание |
|---|---|---|
| align | right, left, center | горизонтальное выравнивание |

| |
|---|
| bgcolor |
| цвет |
| определяет цвет фона блока/контейнера |

| |
|---|
| border |
| числовое значение |
| определяет толщину границ блока/контейнера |
| height |
| числовое значение |
| определяет высоту |
| width |
| числовое значение |
| ширина элемента |
| maxlines |
| числовое значение |
| обрезает текст до N строк |

## Теги <table>, <tr>, <td> и их частные атрибуты

Элемент <table> служит контейнером для элементов, определяющих содержимое таблицы.

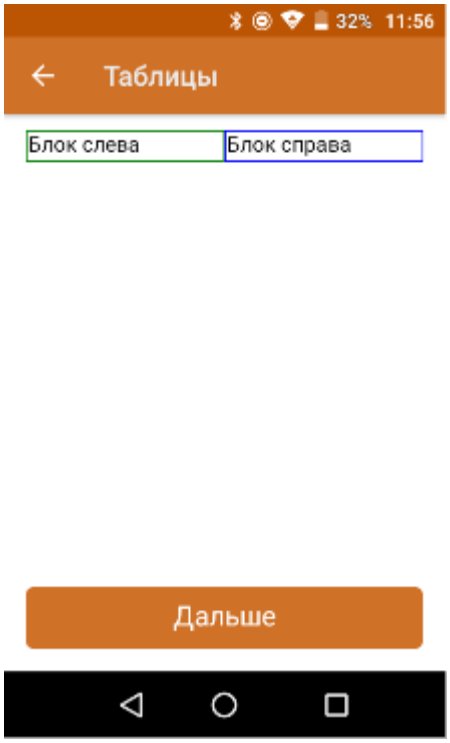
Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <tr> и <td>.

Внутри <table> допустимо использовать такие атрибуты как bgcolor, cellpadding, cols, valign (подробнее см. список ниже).

> Теги <table>, <tr>, <td> обязательно должны иметь завершающий тег.

Тег <td> предназначен для создания одной ячейки таблицы. Данный тег должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

Пример простого синтаксиса:




```
<div>
<table width="100%">
<tr>
<td style="border:1dp solid green;" width="50%">
Блок слева
</td>
<td style="border:1dp solid blue;">
Блок справа
</td>
</tr>
</table>
</div>
```

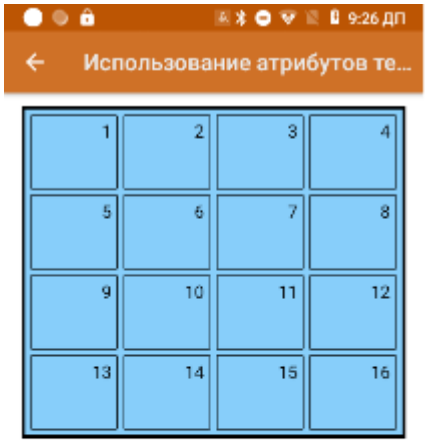
**Для тега <table> доступны следующие частные атрибуты:**

| Атрибут |
| --- |
| Значение |
| Описание |

| |
|---|
| align |
| right, left, center |
| горизонтальное выравнивание |

| |
|---|
| bgcolor |
| цвет |
| определяет цвет фона таблицы |
| border |
| числовое значение |
| определяет толщину границ таблицы, а также включает границы ячеек толщиной 1px |
| height |
| числовое значение |
| определяет высоту таблиц |
| valign |
| top, middle, bottom |
| вертикальное выравнивание содержимого ячеек |
| width |
| числовое значение |
| определяет ширину таблиц |
| cellpadding |
| любое целое значение в пикселах или процентах от доступного пространства |
| определяет отступ содержимого ячейки от границы ячейки |
| cellspacing |
| любое целое положительное число |
| определяет отступ между ячейками |
| cols |
| любое целое положительное число |
| определяет количество колонок |

Пример использования несколько атрибутов сразу:




```
<div width="100%" height="50%">
<table width="100%" cols="4" align="right" valign="top" cellpadding="5dp" cellspacing="5dp" border="3dp"
bgcolor="#87cefa">
<tr>
<td height="25%">1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td height="25%">5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td height="25%">9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td height="25%">13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</table>
</div>
```

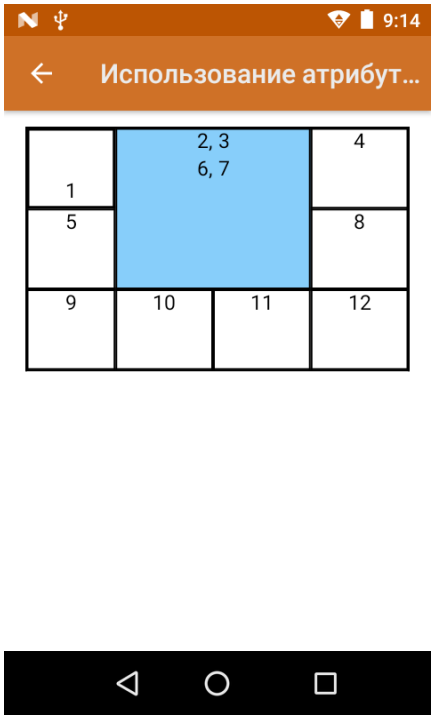
Тег <tr> используется для разметки строки таблицы. Обязательно должен иметь завершающий тег. Данный тег не имеет дополнительных атрибутов.

Тег <td> используется внутри тега <tr> и определяет ячейку таблицы. Обязательно должен иметь завершающий тег.

Для данного тега доступны следующие частные атрибуты:

| Атрибут |
| --- |
| Значение |
| Описание |
| align |
| right, left, center |
| горизонтальное выравнивание в данной ячейке |
| bgcolor |
| цвет |
| определяет цвет фона ячейки |
| border |
| числовое значение |
| определяет толщину границ ячеек |
| colspan |
| любое целое положительное число больше 1 |
| определяет горизонтальное количество ячеек для объединения |
| height |
| числовое значение |
| определяет высоту ячейки |
| rowspan |
| любое целое положительное число больше 1 |
| определяет вертикальное количество ячеек для объединения |
| valign |
| top, middle, bottom |
| вертикальное выравнивание содержимого ячеек |
| width |
| числовое значение |
| определяет ширину ячейки |

Пример использования нескольких атрибутов сразу:



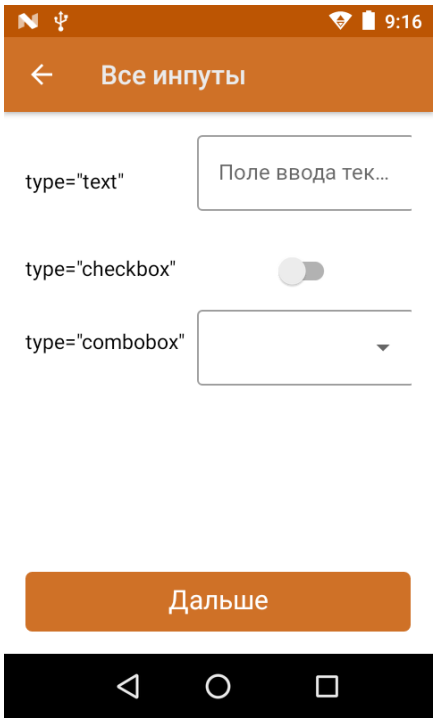

```
<div width="100%">
<table width="100%" border="1dp" align="center">
<tr>
<td height="15%" valign="bottom" border="2dp" bgcolor="#87cefa">1</td>
<td colspan="2" rowspan="2">2, 3<br />6, 7</td>
<td>4</td>
</tr>
<tr>
<td height="15%">5</td>
<td>8</td>
</tr>
<tr>
<td height="15%">9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</table>
</div>
```

## Тег <input> и его частные атрибуты

Тег используется для создания полей ввода текста, чекбоксов или выпадающих списков.

У данного тега есть основной атрибут type с помощью которого определяется, как будет выглядеть поле ввода.

Рассмотрим на следующем примере:




```
<div><div>
<table col="2" width="100%"style="vertical-align:middle;">
<tr>
<td width="30%" height="10%">
type="text"
</td>
<td width="70%" align= "center">
<input type="text" width="90%" tabIndex="1" placeholder="Поле ввода текста" value="{textValue}" datatype="char"
/>
</td>
</tr>
<tr>
<td width="30%" height="10%">
type="checkbox"
</td>
<td width="70%" align= "center">
<input type="checkbox" tabIndex="2" value="{chkbxValue}" onselected ="1" />
</td>
</tr>
<tr>
<td width="30%" height="10%">
type="combobox"
</td>
<td width="70%" align= "center">
<input type="combobox" width="90%" tabIndex="3" placeholder ="пусто" source="СтрокиДляОтображения.Rows"
listItemDisplayTemplate ="{Item.Наименование}" SelectedValue="{listValue}" onselected="Возврат" />
</td>
</tr>
</table>
</div>
```

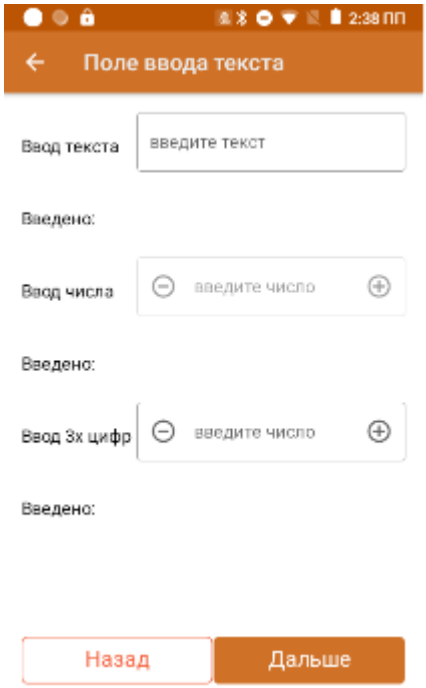
Для создания полей ввода текста доступны следующие частные атрибуты:

| Атрибут | |
|---|---|
| Значение | |
| Описание | |

| |
|---|
| enabled |
| true/false, булевая переменная или выражение |
| отвечает за доступность элемента |

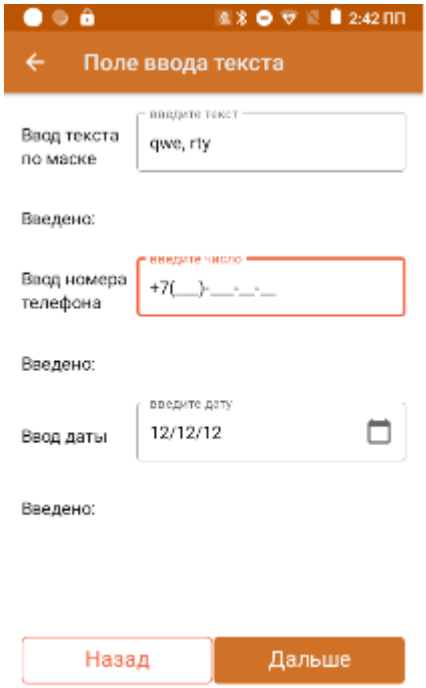
| |
|---|
| placeholder |
| текстовая строка, если внутри строки предполагается пробел, ее необходимо брать в двойные или одинарные кавычки |
| подсказка, отображаемая в тот момент, когда поле ввода пусто (android) |
| datatype |
| decimal, string, datetime |
| тип данных, которые можно вводить |
| height |
| числовое значение |
| определяет высоту поля ввода |
| format |
| типы входных данных в формате HTML |
| принимает regExp, по которому происходит валидация введенных данных |
| width |
| числовое значение |
| определяет ширину поля ввода |
| value |
| имя переменной |
| переменная для занесения вводимой строки<br>value = {var} |
| mask |
| содержит маску для текстовых/числовых данных |
| содержит в себе шаблон, с помощью которого формируются вводимые данные<br>Синтаксис: mask = "##.##" — для ввода чисел<br>mask = "__.__" — для ввода букв и чисел |
| onBlur |
| имя действия |
| указывает действие, на которое будет выполнен переход по смене фокуса (для ОС Android) |
| id/class/style |
| см. Общие атрибуты |
| работают по стандартным правилам HTML |

Пример использования нескольких атрибутов сразу:




```
<div width="100%">
<table cols="2" width="100%" style="vertical-align:middle;">
<tr>
<td width="30%" height="10%" >Ввод текста</td>
<td width="70%" ><input type="text" value="{textValue}" width="100%" onBlur="Возврат" placeholder="введите
текст" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{textValue:(0)}</td>
</tr>
<tr>
<td>Ввод числа</td>
<td>
<input type="text" value="{decimalValue}" enabled="false" width="100%" onblur="Возврат" placeholder="введите
число" datatype="decimal" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{decimalValue:(0)}</td>
</tr>
<tr>
<td>Ввод 3х цифр</td>
<td><input type="text" value="{formatValue}» width="100%" onblur="Возврат" placeholder="введите число"
format="[0-9]{3}" datatype="decimal" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{formatValue:(0)}</td>
</tr>
</table>
</div>
```

Пример использования атрибутов mask и pattern:

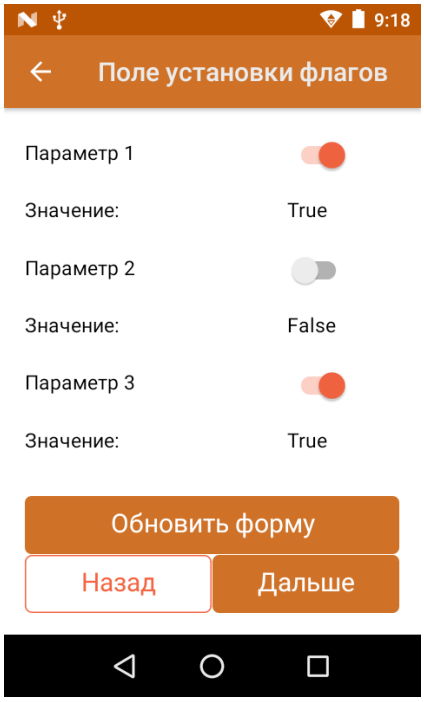



```
<div width="100%">
<table cols="2" width="100%" style="vertical-align:middle;">
<tr>
<td width="30%" height="10%" >Ввод текста по маске</td>
<td width="70%" >
<input type="text" value="{maskValue}" width="100%" onBlur="Возврат" placeholder="введите текст" mask="___,
___" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{maskValue:(0)}</td>
</tr>
<tr>
<td>Ввод номера телефона</td>
<td>
<input type="text" value="{numValue}" width="100%" onblur="Возврат" placeholder="введите число»
mask="+7(###)-###-##-##» /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{numValue:(0)}</td>
</tr>
<tr>
<td>Ввод даты</td>
<td><input type="text" value="{dateValue}" width="100%" onblur="Возврат" placeholder="введите дату"
pattern="dd.MM.yy" datatype="datetime" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{dateValue:(0)}</td>
</tr>
</table>
</div>
```

Для чекбокса доступны следующие частные атрибуты:

| Атрибут |
| --- |
| Значение |
| Описание |
| value |
| для переключателей уникально определяет каждый элемент, с тем, чтобы клиентская или серверная программа могла однозначно установить, какой пункт выбрал пользователь |
| переменная для занесения вводимой строки<br>value = {var} |
| id/class/style |
| см. Общие атрибуты |
| работают по стандартным правилам HTML |

Пример использования нескольких атрибутов сразу:

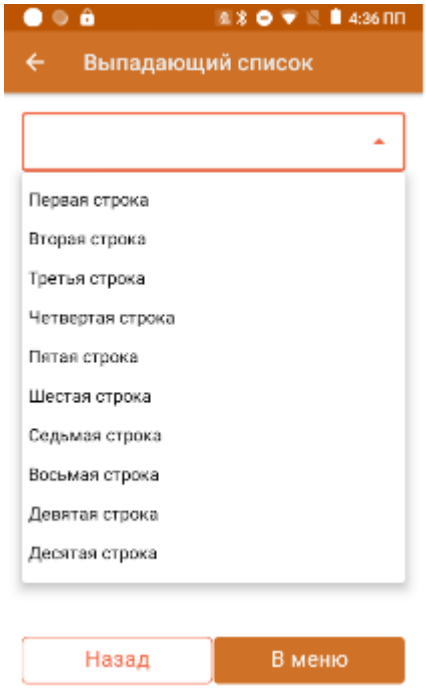



```
<div width="100%" >
<table cols="2" width="100%" style="vertical-align:middle;" >
<tr>
<td width="70%" height="10%">Параметр 1</td>
<td width="30%"><input type="checkbox" value="{par1}" onblur="Возврат" /></td>
</tr>
<tr>
<td width="70%" height="10%">Значение:</td>
<td width="30%">{par1:True;False}</td>
</tr>
<tr>
<td width="70%" height="10%">Параметр 2</td>
<td width="30%"><input type="checkbox" value="{par2}" onblur="Возврат" /></td>
</tr>
<tr>
<td width="70%" height="10%">Значение:</td>
<td width="30%">{par2:True;False}</td>
</tr>
<tr>
<td width="70%" height="10%">Параметр 3</td>
<td width="30%"><input type="checkbox" value="{par3}" onblur="Возврат" /></td>
</tr>
<tr>
<td width="70%" height="10%">Значение:</td>
<td width="30%">{par3:True;False}</td>
</tr>
</table>
</div>
```

Для комбобокса доступны следующие частные атрибуты:

| Атрибут |
| --- |
| Значение |
| Описание |

| |
|---|
| enabled |
| true\|false |
| отвечает за доступность элемента |
| height |
| числовое значение |
| определяет высоту поля ввода |
| width |
| числовое значение |
| определяет ширину поля ввода |
| value |
| является атрибутом без значения |
| атрибут, в котором указывается переменная для занесения выбранной строки<br>Синтаксис: value="{var}" |
| onSelected |
| имя действия |
| указывает действие, на которое будет осуществлен переход после выбора элемента выпадающего списка (для ОС Android) |
| source |
| источник элементов для показа |
| атрибут, который используется для получения коллекции строк.<br>Синтаксис: source = "ItemsCollection". Для того, чтобы передать строки таблицы, необходимо явно (!) указывать строки, т. е.: source = "TableName.Rows" |
| ListItemDisplayTemplate |
| если строка имеет поле ShortName, то будет отображаться оно, если нет — поле Имя |
| позволяет выбрать столбец коллекции для отображения. Синтаксис: ListItemDisplayTemplate = "{Item.ColumnName}" |
| ListItemValueTemplate |
| шаблон вычисления значения поля |
| при выборе позиции списка позволяет произвести вычисление с полем выбранной строки. Синтаксис: ListItemValueTemplate = "{Item.ColumnName = 3+2}" |
| id\|class\|style |
| см. Общие атрибуты |
| работают по стандартным правилам HTML |




```
<div>
<input type="combobox" source="СтрокиДляОтображения.Rows" value ="{SelPos}" listItemDisplayTemplate ="
{Item.Наименование}" listItemValueTemplate="{Item.ВычисляемоеПоле=Item.Код+Item.Характеристика}"
onselected="Возврат" /><br />
{SelPos.Наименование:Выбрано: (0), }{SelPos.Код:(0), }{SelPos.ВычисляемоеПоле:(0)}<br />
</div>
```

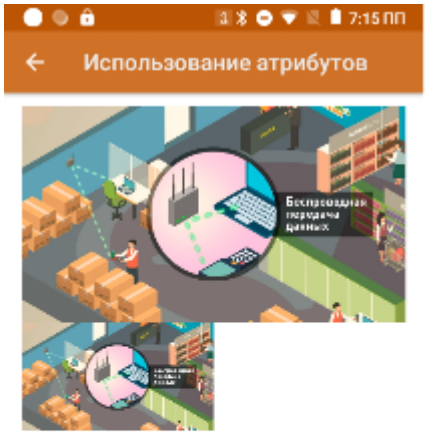
## Тег <img> и его частные атрибуты

Тег используется для отображения изображения. Обязательно должен иметь завершающий тег. С помощью данного тега можно отображать изображения из файла-ресурсника, хранимого в папке «Documents» базы, по ссылке, через переменную из таблицы/по ссылке.

Для отображения изображений доступны следующие частные атрибуты:

| Атрибут | Значение | Описание |
|---|---|---|
| width | числовое значение | определяет ширину изображения |

| |
|---|
| height |
| числовое значение |
| определяет высоту изображения |
| size |
| stretch |
| атрибут, с помощью которого указывается поведение формирования ширины/высоты изображения.<br>В качестве параметра можно указать только stretch (о других неизвестно). Если указано stretch — в случае если изображение больше по ширине или высоте оно будет подогнано под максимальный допустимый размер, ограниченный верстской |
| tcolor |
| цвет |
| |
| id/class/style |
| см. Общие атрибуты |
| работают по стандартным правилам HTML |




```
<div width="100%">
<img size="stretch">bigPict.jpg</img>
<img width="50%" height="30%">bigPict.jpg</img>
</div>
```

## Тег <button> и его частные атрибуты

Тег используется для создания кнопки. Обязательно должен иметь завершающий тег. С помощью данного тега можно создать дополнительные кнопки управления в различных частях окон (в верхней/нижней части/в списке/в самом окне/etc).

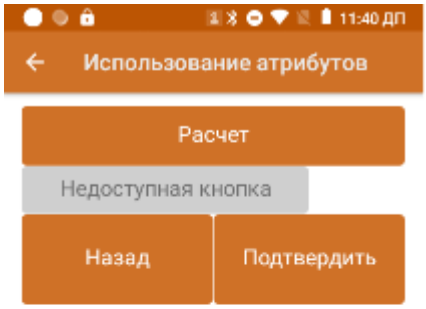
Единственное визуальное действие. в котором нельзя использовать данный тег — действие «Меню», так как оно само состоит из кнопок.

Для создания полей ввода текста доступны следующие частные атрибуты:

| Атрибут |
| --- |
| Значение |
| Описание |

| |
| --- |
| width |
| числовое значение |
| определяет ширину поля ввода |
| height |
| числовое значение |
| определяет высоту поля ввода |
| enabled |
| нет |
| указывает, доступна кнопка для нажатия или нет |
| direction |
| указывается название действия для перехода или:<br>cancel — отмена действия<br>finishproc — возврат на одно действие<br>return — завершить операцию<br>abort — прервать операцию |
| указывает, переход на какое действие будет совершен при нажатии кнопки |
| visible |
| отображает элемент как видимый |
| указывает, отображать кнопку или нет |
| type="submit" |
| кнопка для отправки данных формы на сервер |
| означает, что по нажатию кнопки будет обработаны введенные в поля ввода данные и произведен переход на следующее действие. Раньше, для выполнения подобного действия использовался атрибут direction="ok" |
| id\|class\|style |
| см. Общие атрибуты |
| работают по стандартным правилам HTML |
| command |
| Пример:<br>command="x = y + 1" |
| позволяет производить простые вычисления без перехода на другое действие |

Пример использования атрибутов кнопок:

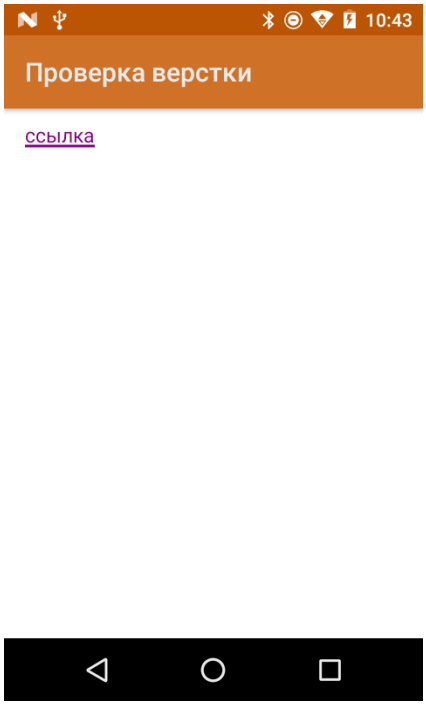



```
<div height="100%" width="100%">
<button width="100%" height="10%" command="value=2+3">Расчет</button><br />
<button width="75%" enabled="false">Недоступная кнопка</button><br />
<button width="100%" visible="false">Невидимая кнопка</button><br />
<button width="50%" height="15%" direction="finishproc">Назад</button><button width="50%" height="15%"
direction="Далее" type="submit">Подтвердить</button>
</div>
```

## Тег <a> и его частные атрибуты

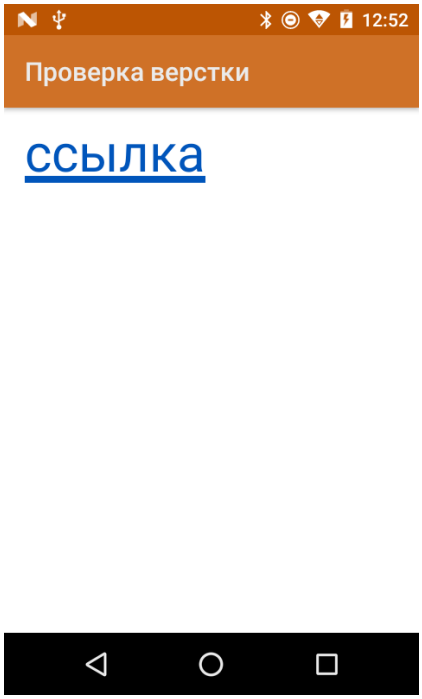
Тег используется для создания гиперссылки на другое действие в алгоритме. Обязательно должен иметь закрывающий тег. Сама ссылка определяется через атрибут href. С помощью атрибута style можно задать цвет гиперссылки.

Пример синтаксиса:




<a href="//www.cleverence.ru/support/2565/Название операции» style="color:#800080">ссылка</a>

Цвет можно задавать как через hex коды цветов, так и по названию цвета.




style="color:purple"

Атрибут size задаёт размер текста гиперссылки

<a href="//www.cleverence.ru/support/2565/Название операции" size="18">ссылка</a>

Для создания ссылки можно вместо атрибута href использовать атрибут direction

<a direction="Название операции">ссылка</a>

форматирование

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Использование классов стилей в Mobile SMARTS

Последние изменения: 2024-03-26

Mobile SMARTS поддерживает работу со многими ТСД, у которых разные размеры и расширения экрана. Для простоты настройки отображения текстов и кнопок для конкретного мобильного устройства предусмотрен стиль клиентского приложения.

> Для мобильных устройств на базе ОС Windows CE интерфейс форматируется согласно инструкции «Стиль клиентского приложения на ТСД».

## Создание классов стилей

Для того, чтобы каждый раз не писать одинаковые параметры, у множества элементов необходимо использовать уже адаптированные стили.

У данного подхода несколько плюсов:

- класс стиля находится в одном месте и если необходимо изменить, например, цвет или размер шрифта, сделать это можно сразу везде без вмешательства в код верстки;
- для устройств на разных ОС (Win/ CE/ Android) можно задавать разные параметры для одних и тех же классов. Это может быть полезно когда разница в разрешениях экранов слишком велика;
- упрощение читаемости html кода.

Файлы стилей должны располагаться на сервере в папке «Documents» (приведем возможные имена файлов):

- **global.css** — общий файл стилей;
- **global.android.css** — файл стилей для Android-устройств;
- **global.win.css** — файл стилей для Win-клиента;
- **global.cf.css** — файл стилей для CE устройств.

Стили из основного файла используются, если не найдены стили из файла для конкретной ОС.

Рассмотрим код класса стиля на следующем примере:

```
.listHead — наименование класса
   {
   text-align: left; - указание, что текст будет прижат к левому краю элемента
   vertical-align: middle; - вертикально посередине элемента
   font-weight: bold; - текст жирный
   font-size: 110%; - размер текста 110%
   color: #666666; - цвет серый
   padding-bottom: 10dp; - снизу элемента отступ внутри элемента 10dp
   }
```

Список свойств достаточно большой, чтобы упомянуть их все в нашей статье, поэтому свойства можно посмотреть на сайте w3schools (поддерживаются не все свойства).

Также рекомендуется ознакомиться с CSS селекторами — это шаблоны, используемые для выбора элементов, которые вы хотите стилизовать (:active|:hover|:disabled используются для кнопок).

Пример использования селекторов для кнопок:

```
<h3 align="center">Нормальное состояние</r>

<button direction="Возврат» width="100%» height="10%» class="bttn3">Кнопка</button>

<h3 align="center">Состояние Active (нажата)</r>

<button direction="Возврат» width="100%» height="10%» class="bttn3">Кнопка</button>

<h3 align="center">Состояние Disabled (недоступна)</r>

<button direction="Возврат» width="100%» height="10%» enabled="false» class="bttn3">Кнопка</button>
```

## Использование цветов в стилях приложения

Начиная с версии платформы Mobile SMARTS 3.3 появилась возможность использовать стили приложения.

В классах используется не конкретный цвет, а ссылка на этот цвет. Способ задания выглядит следующим образом:

```
.some_class{
    color: var(--theme-textColorPrimary);
}
```

Аналогично, можно применять прямо в верстке с помощью style.

Список использующихся стандартных цветов:

| Цвет темы |
| --- |
| **Использование** |
| --theme-colorPrimary |
| Отображение кнопок по умолчанию (цвет фона кнопки обмена в главном меню и кнопки с заливкой, содержимого текстовой и обведенной кнопок, рамки обведенной кнопки) |
| --theme-colorPrimaryOp24 |
| Цвет фона кнопки обмена и кнопки с заливкой при нажатии |
| --theme-colorPrimaryOp12 |
| Цвет фона текстовой и обведенной кнопок при нажатии |
| --theme-colorDivider |
| Цвет разделяющей черты между кнопками в действии «Меню» |
| --theme-colorError |
| Используется в стилях отображения ошибок |
| --theme-colorListItemBackgroundFocused |
| Цвет фона кнопки в фокусе в действии «Меню» |
| --theme-colorListItemBackgroundPushed |
| Цвет фона кнопки при нажатии в действии «Меню» |
| --theme-textColorPrimary |
| Цвет текста по умолчанию |
| --theme-textColorSecondary |
| Вспомогательный цвет текста, используется в стилях текста описаний и комментариев |

> Пользователь не может самостоятельно сменить значения предустановленных цветов.

## Не нашли что искали?

Задать вопрос в техническую поддержку

# Стандартные классы стилей в Mobile SMARTS

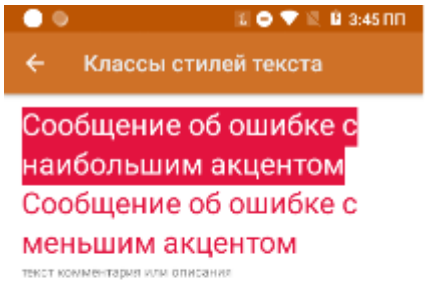
Последние изменения: 2024-03-26

Начиная с версии 3.3 платформы Mobile SMARTS в Android-клиент по умолчанию уже встроены стандартные классы стилей. Использование этих классов обязательно там, где это возможно. Не стоит писать свои аналогичные классы стилей.

**Доступные стили текста:**

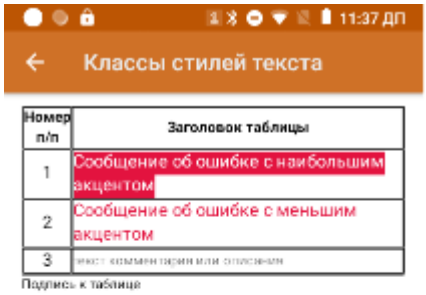
| Класс |
|---|
| Описание |
| .__header_sm_text |
| текст для заголовков (скорее всего для заголовков колонок таблиц) |
| .__error_text |
| отображение уведомлений об ошибке |
| .__error_block |
| отображение уведомлений об ошибке в блоке с красным фоном |
| .__helper_text |
| текст для описаний, комментариев |
| .__caption_text |
| текст для подписей к таблицам, изображениям |

**Пример 1**




```
<r class="__error_block" size="12">Сообщение об ошибке с наибольшим акцентом</r>
<r class="__error_text" size="12">Сообщение об ошибке с меньшим акцентом</r>
<r class="__helper_text" size="12">текст комментария или описания</r>
```

**Пример 2**



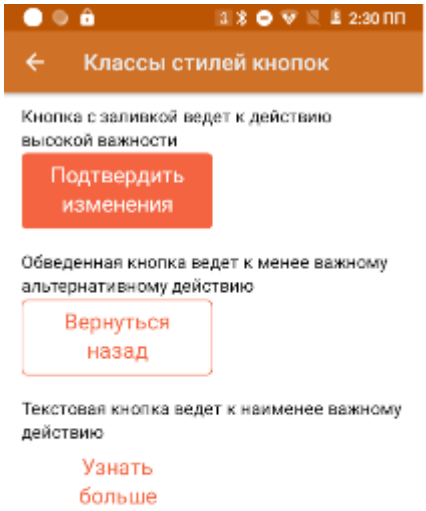

```
<div width="100%">
<table border="1dp" valign="middle">
<tr>
<td width="10%" align="center"><r class="__header_sm_text">Номер п/п</r></td>
<td align="center"><r class="__header_sm_text">Заголовок таблицы</r></td>
</tr>
<tr>
<td align="center">1</td>
<td><r class="__error_block">Сообщение об ошибке с наибольшим акцентом</r></td>
</tr>
<tr>
<td align="center">2</td>
<td><r class="__error_text">Сообщение об ошибке с меньшим акцентом</r></td>
</tr>
<tr>
<td align="center">3</td>
<td><r class="__helper_text">текст комментария или описания</r></td>
</tr>
</table>
<r class="__caption_text">Подпись к таблице</r>
</div>
```

**Доступные стили кнопок:**

| Класс | Описание |
|---|---|
| .__contained_button | кнопки с заливкой используются для наиболее важных действий |
| .__outlined_button | обведенные кнопки используются для менее важных действий |
| .__text_button | текстовые кнопки используются для отображения наименее важных действий |

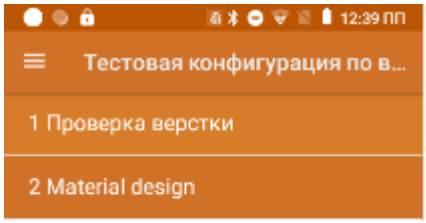
| |
|---|
| .__menu_button |
| кнопка меню в действии «Меню» |
| .__exchange_button |
| кнопка обмена с сервером в главном меню |
| .__exchange_noconn_button |
| кнопка обмена в главном меню при отсутствии соединения с сервером |

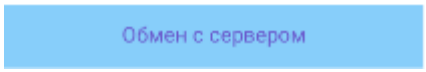
**Пример**




```
Кнопка с заливкой ведет к действию высокой важности
<button class="__contained_button" width="50%">Подтвердить изменения</button>
<br />
Обведенная кнопка ведет к менее важному альтернативному действию
<button class="__outlined_button" width="50%">Вернуться назад</button>
<br />
Текстовая кнопка ведет к наименее важному действию
<button class="__text_button" width="50%">Узнать больше</button>
```

Для изменения стиля кнопки обмена достаточно в файле global.css прописать свои значения атрибутам соответствующего класса, например:




```
.__exchange_button{
background-color: lightskyblue;
color: slateblue;
}
```

То же можно сделать и для других стандартных классов.

Кроме стандартных, можно использовать пользовательские классы стилей. Для их создания используется тег <style>. У каждого стиля есть имя и он хранит в себе параметры key-value.

```
<style name="MyStyle">
<item name="key">value</item>
 </style>
```

## Не нашли что искали?

Задать вопрос в техническую поддержку