

# Дизайнер приложений для MATLAB<sup>®</sup>

## Руководство пользователя

R2014b— Tech Preview

---

Страница 2

### Как связаться с MathWorks

Последние новости: [www.mathworks.com](http://www.mathworks.com)  
Продажи и услуги: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
Сообщество пользователей: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Техподдержка: [www.mathworks.com/support/contact\\_us](http://www.mathworks.com/support/contact_us)

Телефон: 508-647-7000

MathWorks, Inc.  
3 Apple Hill Drive  
Natick, MA 01760-2098

App Designer для MATLAB ® Руководство пользователя

© АВТОРСКОЕ ПРАВО 2014, MathWorks, Inc.

Программное обеспечение, описанное в этом документе, поставляется с лицензионным соглашением. Программное обеспечение может быть использовано или копируется только в соответствии с условиями лицензионного соглашения. Никакая часть этого руководства не может быть фотокопирована или воспроизведена в любой форме без предварительного письменного согласия The MathWorks, Inc.

ФЕДЕРАЛЬНОЕ ПРИОБРЕТЕНИЕ: Это положение применяется ко всем приобретениям Программы и Документации. через, за или через федеральное правительство Соединенных Штатов. Принимая доставку Программы или Документации, правительство настоящим соглашается, что это программное обеспечение или документация квалифицируется как коммерческое программное обеспечение или документация по коммерческому компьютерному программному обеспечению в качестве используемых терминов или определено в FAR 12.212, DFARS Part 227.72 и DFARS 252.227-7014. Соответственно, условия и условия настоящего Соглашения и только те права, которые указаны в настоящем Соглашении, принадлежат и регулировать использование, модификацию, воспроизведение, выпуск, исполнение, отображение и раскрытие Программы и Документации федеральным правительством (или другим юридическим лицом, приобретающим для или через федеральный правительство) и заменяет любые противоречащие договорные условия. Если эта лицензия не работает для удовлетворения потребностей правительства или не согласуется в каком-либо отношении с федеральным законом о закупках, Правительство обязуется вернуть Программу и Документацию, неиспользованные, The MathWorks, Inc.

Торговая марка  
MATLAB и Simulink являются зарегистрированными товарными знаками The MathWorks, Inc.

[www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) для списка дополнительных торговых марок. Другой продукт или бренд названия могут быть товарными знаками или зарегистрированными товарными знаками соответствующих владельцев.

Патенты  
Продукты MathWorks защищены одним или несколькими патентами США. Посмотрите пожалуйста [www.mathworks.com/patents](http://www.mathworks.com/patents) для получения дополнительной информации.

лист регистраций изменений  
Октябрь 2014                      Только онлайн                      Новое в MATLAB версии 8.4 (выпуск 2014b  
- Tech Preview)

Основы дизайнера приложений

1

<a href="#">Откройте App Designer</a> . . . . .	<a href="#">1-2</a>
<a href="#">Создание и модификация приложения</a> . . . . .	<a href="#">1-3</a>
<a href="#">Создать и запустить простое приложение</a> . . . . .	<a href="#">1-3</a>
<a href="#">Запустите существующее приложение</a> . . . . .	<a href="#">1-4</a>
<a href="#">Редактировать существующее приложение</a> . . . . .	<a href="#">1-5</a>
<a href="#">Удалить App Designer Tech Preview</a> . . . . .	<a href="#">1-6</a>

Макет приложения

2

<a href="#">Добавить и удалить компоненты</a> . . . . .	<a href="#">2-2</a>
<a href="#">Выбор компонентов</a> . . . . .	<a href="#">2-3</a>
<a href="#">Изменить размер окна приложения</a> . . . . .	<a href="#">2-4</a>
<a href="#">Переместить компоненты</a> . . . . .	<a href="#">2-5</a>

<a href="#">Выровнять компоненты</a> .....	<a href="#">2-6</a>
<a href="#">Ручное выравнивание компонентов</a> .....	<a href="#">2-6</a>
<a href="#">Автоматическое выравнивание компонентов</a> .....	<a href="#">2-6</a>
 <a href="#">Космические компоненты</a> .....	 <a href="#">2-7</a>

III

[Программирование приложений](#)

3

<a href="#">Генератор кода в App Designer</a> .....	<a href="#">3-2</a>
 <a href="#">Имя компонента в коде приложения</a> .....	 <a href="#">3-5</a>
 <a href="#">Обратные вызовы - запрограммированный ответ на действия пользователя</a> .....	 <a href="#">3-6</a>
<a href="#">Создание обратного вызова</a> .....	<a href="#">3-6</a>
<a href="#">Инициализация приложения</a> .....	<a href="#">3-8</a>
 <a href="#">Отладка кода приложения</a> .....	 <a href="#">3-10</a>
 <a href="#">Обмен данными через обратные вызовы</a> .....	 <a href="#">3-11</a>
 <a href="#">Функции приложения</a> .....	 <a href="#">3-13</a>
 <a href="#">Переименовать компоненты, свойства, обратные вызовы и функции</a> .....	 <a href="#">3-16</a>
<a href="#">Переименовать элементы из представления кода</a> .....	<a href="#">3-16</a>
<a href="#">Переименование компонентов в режиме конструктора</a> .....	<a href="#">3-16</a>
 <a href="#">Удалить компоненты, свойства, обратные вызовы и функции</a> .....	 <a href="#">3-18</a>
<a href="#">Удалить элементы из представления кода</a> .....	<a href="#">3-18</a>
<a href="#">Удалить компоненты из представления конструктора</a> .....	<a href="#">3-18</a>
 <a href="#">Примеры приложений</a> .....	 <a href="#">3-19</a>
<a href="#">Приложение, отображающее один участок по осям</a> .....	<a href="#">3-19</a>
<a href="#">Приложение, отображающее несколько осей</a> .....	<a href="#">3-22</a>
<a href="#">Приложение, отображающее два графика в одной оси</a> .....	<a href="#">3-25</a>
<a href="#">Приложение отображает значение ползунка по мере его изменения</a> .....	<a href="#">3-27</a>
<a href="#">Приложение для изменения цвета лампы на основе выбора группы кнопок</a> .....	<a href="#">3-29</a>
<a href="#">Приложение, которое изменяет размер компонента при изменении размера окна</a> .....	<a href="#">3-30</a>

[Горячие клавиши](#)

4

<a href="#">Ярлыки выбора компонентов</a> .....	<a href="#">4-2</a>
---	---------------------

<a href="#">Ярлыки перемещения компонентов</a> .....	<a href="#">4-3</a>
 <a href="#">Ярлыки изменения размера компонентов</a> .....	 <a href="#">4-4</a>
 <a href="#">Компонент Копировать, Вставить, Дублировать и Удалить ярлыки</a> ..	 <a href="#">4-5</a>

<a href="#">Ярлыки выравнивания компонентов</a> .....	<a href="#">4-6</a>
<a href="#">Восстановить, отменить и отменить ярлыки операций</a> .....	<a href="#">4-7</a>
<a href="#">Запускать, сохранять и открывать ярлыки приложений</a> .....	<a href="#">4-8</a>

[Ссылка на недвижимость](#)

[5](#)

[Справочник по функциям](#)

[6](#)

v

vi

---

Страница 7

1

## Основы дизайнера приложений

- « [Открыть конструктор приложений](#)» на стр. 1-2.
- « [Создание и изменение приложения](#)» на стр. 1-3.
- « [Удаление App Designer Tech Preview](#)» на стр. 1-6.

---

Страница 8

1 Основы дизайнера приложений

## Открыть дизайнер приложений

В командном окне MATLAB ® введите:

```
appdesigner
```

Чтобы открыть ранее созданное приложение с помощью дизайнера приложений, введите следующее, где *имя файла* — имя файла приложения (.mlapp) в пути поиска MATLAB:

```
appdesigner имя файла
```

1-2

---

Страница 9

Создание и модификация приложения

## Создание и модификация приложения

В этой секции...

[«Создание и запуск простого приложения» на стр. 1-3](#)

[«Запустите существующее приложение» на стр. 1-4](#)

[«Редактировать существующее приложение» на стр. 1-5](#)

### Создать и запустить простое приложение

В этом примере показано, как создать и запустить простое приложение. Приложение отображает значение ползунок после того, как пользователь перемещает ползунок в новое место.

- 1 Откройте Дизайнер приложений.  
В командном окне MATLAB введите appdesigner.
- 2 Из палитры компонентов App Designer слева перетащите ползунок и цифру.  
Редактировать поле в центральной области дизайна.

1-3

## Страница 10

### 1 Основы дизайнера приложений

- 3 В центральной области дизайна щелкните правой кнопкой мыши ползунок и выберите « **Обратные вызовы** » > « **Редактировать** ».  
**ValueChangedFcn** .
- 4 В диалоговом окне «Редактировать обратный вызов ValueChangedFcn» примите имя по умолчанию нажав **ОК** .

Хотя это и не сделано здесь (для простоты примера), изменение имени функции облегчает связывание кода обратного вызова с компонентом, особенно когда в приложении есть несколько экземпляров одного и того же компонента.

- 5 В открывшемся файле кода добавьте следующий код под функцию  
SliderValueChanged (приложение). Код указывает MATLAB обновить числовое редактирование поле для отображения положения ползунка.

```
app.NumericEditField.Value = app.Slider.Value;
```

Ваш код может получить доступ ко всем компонентам и их значениям напрямую, подобно тому, как что вы обращаетесь к полям массива структуры. Для доступа или установки значения свойства укажите приложение, которое вы разрабатываете (которое всегда является приложением), компонент и имя свойства компонента.

- 6 На вкладке « **Редактор** » нажмите « **Запустить приложение** » .

Сохраните приложение при появлении запроса.

Протестируйте приложение, изменив значение ползунка и отпустив кнопку мыши. Числовой поле редактирования отражает новое значение.

[«Приложение, отображающее значение ползунка по мере его изменения» на странице 3-27](#), обновляет поле редактирования непрерывно, как ползунок перемещается.

### Запустите существующее приложение

Чтобы запустить приложение, которое вы создали ранее:

- 1 Откройте дизайнер приложений:  
appdesigner
- 2 На вкладке « **Дизайнер** » нажмите « **Открыть** » .
- 3 В диалоговом окне Открыть перейдите к папке, содержащей файл MLAPP, и укажите имя в поле **Имя файла** .
- 4 Нажмите **Открыть** .

1-4

---

Страница 11

Создание и модификация приложения

App Designer отображает приложение в режиме конструктора.

- 5 На вкладке « **Дизайнер** » нажмите « **Выполнить** » .

Кроме того, вы можете ввести имя файла ( *исключая* расширение файла .mlapp) в Командное окно MATLAB. Файл должен находиться в папке на вашем пути MATLAB.

### Изменить существующее приложение

Чтобы открыть App Designer и отредактировать существующее приложение:

- 1 Открыть дизайнер приложений  
  
`appdesigner`
- 2 На вкладке « **Дизайнер** » нажмите « **Открыть** » .
- 3 В диалоговом окне Открыть перейдите к папке, содержащей файл MLAPP, и укажите имя в поле **Имя файла** .
- 4 Нажмите **Открыть** .

App Designer отображает приложение в режиме конструктора.

- 5 Нажмите « **Вид кода** » для просмотра и редактирования кода приложения.

В качестве альтернативы, в командном окне MATLAB введите эту команду, где *filespec* - это имя файла, полный путь к файлу или частичный путь к файлу приложения (.mlapp):

спецификатор *файла приложения*

Если вы введете только имя файла, файл должен находиться в папке на вашем пути MATLAB.

1-5

---

Страница 12

- 1 Основы дизайнера приложений

## Удаление App Designer Tech Preview

Чтобы удалить программное обеспечение предварительного просмотра App Designer и документацию, введите команда в командном окне MATLAB:

`uninstallappdesigner`



1-6

---

Страница 13

2

## Макет приложения

- « [Добавить и удалить компоненты](#)» на стр. 2-2
- « [Выбор компонентов](#)» на стр. 2-3
- « [Изменить размер окна приложения](#)» на стр. 2-4
- « [Переместить компоненты](#)» на стр. 2-5
- « [Выравнивание компонентов](#)» на стр. 2-6.
- « [Космические компоненты](#)» на стр. 2-7

## Добавить и удалить компоненты

Когда вы впервые открываете App Designer, область дизайна, представляющая ваше приложение, пуста. Чтобы заполнить его компонентами, перетащите компоненты из палитры компонентов (на слева) в зону дизайна (посередине).

Чтобы добавить или удалить компоненты, вы можете:

- Вырезать, копировать, вставлять или удалять один компонент в области дизайна.

Щелкните правой кнопкой мыши компонент и выберите параметр в контекстном меню.

- Вырезать, копировать, вставлять или удалять несколько компонентов в области дизайна.

Сделайте множественный выбор компонентов, щелкните правой кнопкой мыши и выберите вариант из контекста меню.

- Дублируйте компонент.

Щелкните правой кнопкой мыши компонент и перетащите дубликат на новое место в текущем приложении.

Примечание. При дублировании или копировании и вставке компонентов новые компоненты те же значения свойств, что и у исходных компонентов, за исключением обратных вызовов,

**Имя** в свойствах **кода** и **позиции** . Эти свойства установлены по умолчанию ценности.

Вы можете копировать или вырезать компоненты из одного приложения и вставлять их в другое.

2-2

Выбор компонентов

## Выбор компонентов

Выбор компонентов - это первый шаг при выполнении множества операций.

выбор

Один компонент

Несколько компонентов

Как

Нажмите на компонент.

Удерживая клавишу **Shift** , нажимая каждый компонент, который вы хотите включить в выбор.

Или же

Нажмите на пустую область в области дизайна, а затем перетащите границу выделения вокруг все компоненты, которые вы хотите выбрать. App Designer выбирает только компоненты полностью содержится в выборе.

Все выбираемые компоненты, кроме это дети другого компонента.

Нажмите **Ctrl + A** .

Отменить выбор ранее выбранного компонента или повторно выбрать невыбранный компонент выбор).

Нажмите **Shift + клик** .

Чтобы переместить выделение на следующий компонент, контейнер или окно приложения в дизайне последовательность навигации по клавише **Tab** , нажмите клавишу **Tab** . Чтобы переместить выделение в предыдущий компонент, контейнер или окно приложения в последовательности навигации по клавише табуляции, нажмите **Shift + Tab** .

2-3

---

## Страница 16

2 Макет приложения

### Изменить размер окна приложения

Чтобы изменить размер окна приложения, щелкните его границу в области дизайна, а затем перетащите расширитель.

Чтобы сохранить текущее соотношение сторон, поместите курсор на расширитель. Удерживать Клавиша **Shift** при перетаскивании расширителя.

Чтобы отменить операцию изменения размера в процессе, нажмите клавишу **Escape** .

2-4

---

## Страница 17

Переместить компоненты

### Переместить компоненты

Вы можете перемещать компоненты вручную (от руки) или с установленными приращениями следующим образом:

- От руки - перетащите компонент или компоненты.
- Шаг в 1 пиксель - выберите компонент или компоненты. Нажмите **стрелку** на клавиатуре введите направление, в котором вы хотите переместить компонент.
- 10-пиксельные приращения - выберите компонент или компоненты. Нажмите **Shift + стрелка** в направление, в котором вы хотите переместить компонент.

Чтобы отменить операцию ручного перемещения, нажмите клавишу **Escape** .

2-5

---

## Страница 18

2 Макет приложения

### Выровнять компоненты

Вы можете указать App Designer автоматически выравнивать компоненты или выровнять их вручную. Например, вы можете использовать ручной метод для первоначального макета, а затем

отполировать макет с помощью автоматического выравнивания.

### Ручное выравнивание компонентов

При перетаскивании компонентов в области дизайна вы увидите подсказки по выравниванию, которые помогут вам в расстановке компонентов.

- Пунктирные оранжевые линии указывают, что компоненты выровнены по центру.
- Сплошные оранжевые линии указывают, что компоненты выровнены по их краям (вверху, дно или бока).

Советы по выравниванию не отображаются, если выбрать компонент, а затем нажать клавиши со стрелками для переместить компонент.

### Автоматическое выравнивание компонентов

Чтобы использовать функцию автоматического выравнивания:

- 1 В области дизайна выберите два или более компонентов.
- 2 На вкладке « **Макет** » в разделе « **Выравнивание** » щелкните элемент управления, чтобы указать, как хочу, чтобы эти компоненты выровнялись.

2-6

## Космические Компоненты

Вы можете настроить App Designer для автоматического размещения компонентов или вручную укажите интервал следующим образом:

- 1 В области дизайна выберите компоненты, которые вы хотите разместить.
 

Вы должны выбрать два или более компонентов для пробела вручную и три или более компоненты в космос автоматически.
- 2 На вкладке « **Макет** » в разделе « **Интервал** » выполните одно из следующих действий:
  - Для ручного интервала выберите **Использовать .... пиксели** и введите значение в поле **пикселей** .
  - Для автоматического пробела выберите « **Авто пробел** » .

Autospace для дизайнера приложений:

- Равномерное распределение компонентов между верхним и нижним компонентами когда вы применяете вертикальный интервал.
- Равномерное распределение компонентов между самым левым и самым правым компоненты, когда вы применяете горизонтальный интервал.

Иногда количество пикселей между двумя компонентами мешает приложению Дизайнер делает пространство между каждым компонентом идентичным. Однако Разница в расстоянии не более 1 пикселя.

- 3 Выберите способ применения интервала, выбрав **Применить горизонтально** или **Применить вертикально** .

2-7

---

Страница 20

2-8

---

Страница 21

3

# Программирование приложений

- « [Генерация кода в дизайнера приложений](#)» на стр. 3-2
- « [Имя компонента в коде приложения](#)» на стр. 3-5.
- « [Обратные вызовы - запрограммированный ответ на действия пользователя](#)» на стр. 3-6.
- « [Отладка кода приложения](#)» на стр. 3-10
- « [Обмен данными через обратные вызовы](#)» на стр. 3-11.
- « [Функции приложения](#)» на стр. 3-13
- « [Переименование компонентов, свойств, обратных вызовов и функций](#)» на стр. 3-16.
- « [Удалить компоненты, свойства, обратные вызовы и функции](#)» на стр. 3-18.
- « [Примеры приложений](#)» на стр. 3-19

---

## Страница 22

### 3 Программирование приложений

## Генератор кода в дизайнера приложений

Когда вы создаете приложение, App Designer генерирует код, отражающий макет вашего компонента и настройки. Кроме того, он генерирует структуру для функций обратного вызова, который контролирует поведение приложения. Когда вы сохраняете свое приложение, App Designer сохраняет код в Файл MLAPP. Когда MATLAB запускает ваше приложение, оно запускает этот файл.

App Designer автоматически генерирует и управляет кодом для создания приложения, его компонентов, и рамки для функций. Чтобы вы случайно не перезаписали код, которым управляет App Designer, этот код только для чтения (обозначается светло-серым фон).

Код, который автоматически генерирует App Designer, является модульным. Например, когда вы добавляете кнопка и поле редактирования в области дизайна сгенерированного кода, как показано на рисунке что следует.

3-2

---

**Страница 23**

Генератор кода в дизайнере приложений

3-3



Заметить, что:

- Первый раздел кода определяет свойства приложения.

Первоначально этот раздел содержит свойства, которые соответствуют приложению и приложению. компоненты. Если вы добавляете свойства для обмена данными, App Designer добавляет ваши свойства в этот регион. Для получения информации о добавлении свойств см. [«Обмен данными через обратные вызовы» на странице 3-11](#)

- Второй раздел кода содержит функции приложения.

Первоначально этот раздел содержит основу для функции запуска. Если вы добавите обратные вызовы или другие функции приложения, дизайнер приложений добавляет их в этот регион. За информацию о добавлении функций см. в разделе [«Обратные вызовы - запрограммированный ответ пользователю» Действия» на стр. 3-6](#) и «Функ

- Третий раздел кода инициализиру

Если вы выбираете компонент в режиме конструктора и используете вкладку **Свойства**, чтобы изменить значения свойств для этого компонента, App Designer отражает эти изменения в этом разделе кода.

Если вы переименуете компонент, свойство или функцию (включая функцию обратного вызова), приложение Дизайнер обновляет имя в областях кода только для чтения и чтения и записи.

3-4

Имя компонента в коде приложения

## Имя компонента в коде приложения

App Designer ссылается на компоненты в коде, используя имя компонента. Если там есть более одного компонента, App Designer добавляет имя компонента с числовое значение. Например, если вы включаете три кнопки в свое приложение, App Designer называет их Button, Button2 и Button3.

Лучше всего переименовывать каждый компонент с именем, которое поможет вам его идентифицировать. когда вы читаете код. Например, если кнопка заставляет приложение отображать результаты, Вы можете переименовать его ResultsButton. Если вы создаете приложение и принимаете имена по умолчанию, но позже решите изменить их, вы можете сделать это, как описано в [«Переименовать компоненты. Свойства, обратные вызовы и функции» на стр. 3-16](#).

---

**Страница 26****3** Программирование приложений

## Обратные вызовы - запрограммированный ответ на действия пользователя

Обратный вызов - это функция, которая выполняется в ответ на некоторые заранее определенные действия пользователя, такие как нажатием кнопки или вводом значения в поле ввода чисел.

Большинство компонентов приложения поддерживают обратные вызовы. Метки, оси, переключатели, переключатели, лампы и датчики нет. (Тем не менее, группы кнопок, которые содержат радио или кнопки переключения есть функции обратного вызова.)

### Создание обратного вызова

Вы можете создать обратный вызов компонента из представления конструктора или кода:

- Для создания обратного вызова из поля зрения дизайна, щелкните правой кнопкой мыши компонент и выберите **обратные вызовы** > **Изменить** имя обратного вызова.

- Чтобы создать обратный вызов из представления кода, в браузере кода щелкните правой кнопкой мыши компонент и выберите **обратные вызовы** > **Edit** обратного вызова имя.

---

Страница 27

Обратные вызовы - запрограммированный ответ на действия пользователя

Затем выполните следующие действия:

- 1 В диалоговом окне Add Callback Function введите имя для вашего обратного вызова и нажмите **ОК**.

Дизайнер приложений отображает представление кода.

Рекомендуется назвать обратный вызов, чтобы вы могли легко определить элемент управления с которым это связано в коде приложения и что делает обратный вызов. Например, если вы хотите, чтобы нажатие кнопки приводило к отображению данных, вы можете назвать Кнопка обратного вызова PlotButtonPressed.

- 2 В том месте кода, где дизайнер приложений помещает курсор, введите код, который Вы хотите, чтобы MATLAB запускался, когда пользователь манипулирует элементом управления.

Обратитесь к компонентам и их свойствам с помощью приложения. в качестве префикса для каждого компонента. Это позволяет получить доступ к свойствам напрямую, аналогично тому, как вы поля структуры. Например, чтобы установить свойство Enabled ползунка с именем

Чтобы отключить TempSlider, используйте следующий код:

```
app.TempSlider.Enabled = 'off' ;
```

Совет. Чтобы просмотреть имена свойств компонентов и краткое описание каждого из них, щелкните информацию. значок рядом с компонентом в браузере кода.

3-7

---

Страница 28

3 Программирование приложений

Полные описания свойств доступны в [главе 5](#).

- 3 Запустите ваше приложение.

Убедитесь, что манипулирование компонентом дает желаемый эффект. Если ваше приложение возвращает ошибки, отладьте ваш код, как описано в [разделе «Отладка кода приложения» на стр. 3-10](#).

Примечание. Несколько компонентов не могут совместно использовать обратный вызов. Если вы хотите поделиться кодом среди обратные вызовы, рассмотрите возможность написания функции и вызова ее из нескольких обратных вызовов, как описано в разделе «[Функции приложения» на стр. 3-13](#) .

Подробные примеры обратных вызовов см. В разделе «[Примеры приложений» на стр. 3-19](#) .

## Инициализация приложения

Используйте функцию запуска приложения для выполнения задач, которые вы хотите, чтобы MATLAB выполнял, когда приложение сначала открывается. MATLAB запускает функцию запуска после создания компонентов.

Например, используйте функцию запуска, если вы хотите, чтобы оси отображались с графиком по умолчанию, когда приложение открывается или если вы хотите инициализировать свойства приложения.

Чтобы получить доступ к функции запуска в приложении, в представлении «Код» нажмите « **Просмотреть запуск** ». **Функция** на вкладке « **Редактор** ».

Отображение графика по осям при открытии приложения

В этом примере показано, как создать приложение, которое открывается с графиком по осям.

3-8

В App Designer:

- 1 Перетащите оси в область дизайна.
- 2 Нажмите **Вид кода** .
- 3 На вкладке « **Редактор** » нажмите « **Просмотреть функцию запуска** » .

App Designer помещает курсор в функцию запуска (приложение).

- 4 Добавьте следующий код:

```
x = пробел (0,2 * пи, 100);  
y = грех (x);  
участок (app.Axes, x, y);
```

- 5 Сохраните и запустите приложение.

Когда приложение открывается, приложение отображает оси с указанным графиком.

---

**Страница 30****3** Программирование приложений

## Отладка кода приложения

Многие функции отладки, доступные в редакторе MATLAB, также представлены в App Designer code view. В частности, вы можете:

- Установите и очистите стандартные контрольные точки.
- Проходить по файлу, останавливаясь в тех точках, где вы хотите проверить значения.

После установки одной или нескольких точек останова и нажатия кнопки « **Запустить приложение** » редактор редактора приложений На вкладке отображаются кнопки « **Продолжить** » , « **Шаг** » и « **Отмена** » .

- Просмотр значений переменных в браузере MATLAB Workspace.
- Очистите точку останова, щелкнув по ней.

Подробнее об этих функциях отладки см. В разделе «Процесс и функции отладки». в документации MATLAB.

---

**Страница 31**

## Обмен данными через обратные вызовы

Создайте свойства для хранения данных, которыми вы хотите поделиться через обратные вызовы. Например, вы Можно создать свойство для хранения результата расчета. Свойства префиксов в App Designer что вы создаете с помощью приложения., которые позволяют получить доступ к свойствам напрямую, аналогично способ доступа к полям структурного массива.

Сохранение и отображение количества нажатий кнопок

В этом примере показано, как создать приложение, которое ведет подсчет того, сколько раз кнопка нажата во время сеанса приложения.

- 1 Перетащите **кнопку** и **числовое поле редактирования** в область дизайна.
- 2 Нажмите **Вид кода**.
- 3 В браузере кода в разделе «**Свойства приложения**» нажмите «**Добавить свойство**».
- 4 В открывшемся поле редактирования введите Counter.  
  
Дизайнер приложений добавляет приложение. префикс к счетчику и перечисляет вашу собственность в коде браузер.
- 5 Инициализируйте свойство счетчика.  
а На вкладке «**Редактор** панели инструментов» нажмите «**Просмотреть функцию запуска**».

3-11

## Страница 32

### 3 Программирование приложений

- б В коде в строке указывается следующая функция запуска (приложения):  
  
`app.Counter = 0;`
- 6 В коде браузера, щелкните правой кнопкой мыши **app.Button** и выберите **Callbacks > Edit ButtonPressedFcn**.
- 7 В диалоговом окне Добавить функцию обратного вызова нажмите **ОК**. (Примите обратный вызов по умолчанию имя функции.)
- 8 Добавьте код, чтобы увеличить счетчик и отобразить значение счетчика в числовом редакторе. поле каждый раз, когда пользователь нажимает кнопку.  
  
Сразу после строки функции ButtonButtonPush (app) добавьте:  
  
`app.Counter = app.Counter + 1;  
app.NumericEditField.Value = app.Counter;`
- 9 Сохраните, запустите и протестируйте приложение.

Функции приложения

В дополнение к кодированию функций обратного вызова для вашего приложения вы можете создавать функции приложения которые не зависят от какого-либо конкретного компонента. Например, вы можете создать Функция приложения для выполнения расчета. Затем вы можете вызвать эту функцию из нескольких функции обратного вызова, устраняя необходимость повторять этот код в каждом обратном вызове, который использует расчет.

Приложение для выполнения расчета

В этом примере показано, как создать приложение, которое выполняет вычисления. Он использует функцию, `mysalc`, чтобы вычислить сумму и произведение двух значений, указанных пользователем. Когда пользователь изменяет любое значение, приложение пересчитывает сумму и продукт, вызывая `mysalc`.

В App Designer:

- 1
- Добавьте четыре числовых поля редактирования и метки в область дизайна, как показано в предыдущем изображении и описано в этой таблице.

Составная часть	Ассоциированный ярлык	Имя компонента в коде	Значения свойства по умолчанию
Текст			
Числовое поле редактирования	Значение 1:	Значение1	Никто

Составная часть	Ассоциированный ярлык	Имя компонента в коде	Значения свойства по умолчанию
Текст			

Числовое поле редактирования	Значение 2:	Value2	Никто
Числовое поле редактирования	Сумма:	сумма	<b>Минимум:</b> -Инф
			<b>Максимум:</b> Инф
			<b>Редактируемый:</b> снимите этот флажок коробка.
Числовое поле редактирования	Продукт:	Продукт	<b>Минимум:</b> -Инф
			<b>Максимум:</b> Инф
			<b>Редактируемый:</b> снимите этот флажок коробка.

- 2 Нажмите **Вид кода** .
- 3 Создайте функцию для расчета суммы и произведения значений, которые пользователь вводит для значения 1 и значения 2:
  - a В браузере кода нажмите **Добавить функцию** .
  - б В поле add function введите имя функции, mycalc.
  - с Добавьте код, чтобы получить введенные пользователем значения, выполнить вычисления и назначить вектор к выходному аргументу функции, результат.

Сразу после строки result = mycalc (app) функции добавьте:

```

a = app.Value1.Value;
b = app.Value2.Value;
сумма = a + b;
прод = a * b;
результат = [сумма, продукт];

```
- 4 Закодируйте обратный вызов для поля числового редактирования Value 1, чтобы вызвать функцию mycalc всякий раз, когда пользователь изменяет значение поля:
  - a В коде браузера, щелкните правой кнопкой мыши **app.Value1** и выберите **Callbacks > Edit ValueChangedFcn** .
  - б В диалоговом окне Добавить функцию обратного вызова измените имя на Value1Changed, и затем нажмите **ОК** .

3-14

---

## Страница 35

Функции приложения

- с В коде, начинающемся со строки после функции Value1Changed (app), добавьте следующий код:
 

```

calc = mycalc (приложение);
app.Sum.Value = calc (1);
app.Product.Value = calc (2);

```
  - 5 Запишите обратный вызов для поля редактирования числового значения Value 2, чтобы вызвать функцию mycalc всякий раз, когда пользователь изменяет значение поля:
    - a В коде браузера, щелкните правой кнопкой мыши **app.Value2** и выберите **Callbacks > Edit ValueChangedFcn** .
    - б В диалоговом окне Добавить функцию обратного вызова измените имя на Value2Changed, и затем нажмите **ОК** .
    - с В коде, начинающемся со строки после функции Value2Changed (app), добавьте следующий код:
 

```

calc = mycalc (приложение);
app.Sum.Value = calc (1);
app.Product.Value = calc (2);

```
  - 6 На вкладке « **Редактор** » нажмите « **Запустить приложение** » .
- Сохраните приложение при появлении запроса.

Протестируйте приложение, введя значения для значения 1 и значения 2.



---

**Страница 36**

3 Программирование приложений

## Переименование компонентов, свойств, обратных вызовов и функций

Вы можете переименовать компоненты, свойства и функции (включая обратные вызовы) в вашем приложении. App Designer распространяет изменение имени по всему коду приложения. Ты можешь переименовать любой из этих элементов приложения из представления кода. В режиме дизайна вы можете переименовать только компоненты приложения.

### Переименовать элементы из представления кода

Чтобы переименовать компонент, свойство или функцию из представления кода:

- 1 В браузере кода дважды щелкните имя кода.

- 2 В открывшемся поле редактирования введите новое имя.

App Designer обновляет код для отображения нового имени компонента при нажатии **Введите** или нажмите вдали от поля редактирования.

Вы не можете изменить приложение. префикс. Все имена компонентов приложения должны иметь приложение. префикс.

**Совет** Если вы не уверены, какое имя соответствует определенному компоненту, используйте миниатюру приложения, которое отображается под браузером кода. Нажмите на компонент, что вы хотите идентифицировать. Дизайнер приложений выделяет имя этого компонента в Код браузера.

### Переименование компонентов в режиме конструктора

- 1 В области дизайна выберите компонент, для которого вы хотите изменить код имя.
- 2 Нажмите вкладку **Свойства**.

---

**Страница 37**

В поле « **Имя в коде** » отображается имя, на которое ссылается этот компонент. в коде.

3 В поле **Имя в коде** введите новое имя.

Нажмите **Enter** или щелкните вдали от поля. App Designer обновляет код, чтобы отразить новое имя компонента.

3-17

---

## Страница 38

3 Программирование приложений

### Удалить компоненты, свойства, обратные вызовы и функции

Вы можете удалить компоненты, свойства и функции (включая обратные вызовы) из вашего приложения. Вы можете удалить любой из этих элементов приложения из представления кода. С точки зрения дизайна вы можете удалять только компоненты приложения.

#### Удалить элементы из представления кода

Чтобы удалить компонент, свойство или функцию из представления кода, в браузере кода выберите введите кодовое имя, а затем нажмите клавишу **Delete** .

Дизайнер приложений обновляет код, чтобы удалить удаленный элемент. Если вы удалите компонент, Дизайнер приложений также удаляет его из представления дизайна. Когда вы удаляете компонент, обратные вызовы созданные для этого компонента *не* удаляются. Поэтому код, который вы написали для обратного вызова сохраняется

Совет Если вы не уверены, какое имя соответствует определенному компоненту, используйте миниатюру приложения, которое отображается под браузером кода. Нажмите на компонент что вы хотите идентифицировать. Дизайнер приложений выделяет имя этого компонента в

Код браузера.

### Удалить компоненты из представления конструктора

В области дизайна выберите компонент или компоненты, которые вы хотите удалить, а затем нажмите клавишу **Delete** или **Backspace**.

App Designer обновляет код и область дизайна, чтобы удалить удаленный компонент. Обратные вызовы, созданные для этого компонента, *не* удаляются. Поэтому код, который вы написали для обратный вызов сохраняется.

3-18

---

Страница 39

Примеры приложений

## Примеры приложений

В этой секции...

[«Приложение, отображающее один график по осям» на стр. 3-19](#)

[«Приложение, отображающее несколько осей» на стр. 3-22](#)

[«Приложение, отображающее два графика в одной оси» на стр. 3-25](#)

[«Приложение, отображающее значение ползунка по мере его изменения» на странице 3-27](#)

[«Приложение для изменения цвета лампы на основе выбора группы кнопок» на стр. 3-29](#)

[«Приложение, изменяющее размер компонента при изменении размера окна» на стр. 3-30](#)

### Приложение, отображающее одиночный график по осям

В этом примере показано, как создать приложение для пользователей, чтобы управлять методом построения диаграмм для данные построения. Пользователь выбирает график в раскрывающемся списке.

---

**Страница 40**

3 Программирование приложений

Чтобы создать приложение:

- 1 В конструкторе приложений перетащите оси и раскрывающийся список в центральную область дизайна. организовать их, как показано на предыдущем изображении.
- 2 В области дизайна выберите раскрывающийся список, а затем на вкладке **Свойства** щелкните **Параметры** .
- 3 Установите свойства для раскрывающегося списка, как показано на рисунке ниже.

3-20

---

**Страница 41**

Примеры приложений

Чтобы удалить строку, выберите ее и нажмите кнопку «минус».

Пользователь увидит *текстовые* значения в выпадающем списке. Значения *текстовых данных* являются числовыми значениями, связанные с каждой опцией. Возможно, вам будет проще кодировать обратные вызовы, используя функции, которые работают с числовыми значениями, а не со строковыми значениями.

- 4 Закройте диалоговое окно «Параметры» и нажмите « **Вид кода** » .
- 5 Добавьте свойство приложения для хранения данных для вашего приложения.

В браузере кода нажмите **Добавить свойство** , а затем введите Данные в открывшемся поле.

App Designer добавляет имя вместе с приложением. префикс.

- 6 Укажите график, который вы хотите отображать в приложении при первом его открытии, добавив код для функции запуска.

На вкладке « **Редактор** » нажмите « **Просмотреть функцию запуска** » . Добавьте этот код в автозагрузку функция:

```
app.Data = [75 91 105 123,5 131 150 179 203 226 249 281,5];
бар (приложение. оси, приложение. данные);
```

- 7 Создать обратный вызов для выпадающего.

В коде браузера, щелкните правой кнопкой мыши **app.DropDown** и выберите **Callbacks > Edit ValueChangedFcn** .

3-21

## Страница 42

### 3 Программирование приложений

- 8 В диалоговом окне Добавить функцию обратного вызова нажмите **ОК** . (Примите обратный вызов по умолчанию имя функции.)
- 9 В коде, начинающемся на строке после функции DropDownValueChanged (app), добавьте следующий код:

```
% Получить данные о значениях для опции, выбранной пользователем
val = app.DropDown.ValueData;
% График данных с использованием выбранного метода построения диаграмм
если (val == 1)
    бар (app.Data);
elseif (val == 2)
    Площадь (app.Data);
elseif (val == 3)
    пирог (app.Data);
конец
```

- 10 Сохраните и запустите приложение.

На вкладке « **Редактор** » нажмите « **Запустить приложение** » .

Сохраните приложение при появлении запроса.

Протестируйте приложение, попробовав различные элементы управления приложения в запущенном приложении.

## Приложение, отображающее несколько осей

В этом примере показано, как создать приложение, которое принимает входные параметры и графики данные в двух осях. Параметры определяют изменяющийся во времени и изменяющийся по частоте сигнал. Один график отображает данные во временной области. Другой график отображает данные в частотная область.

Чтобы создать приложение:

- 1
- В конструкторе приложений перетащите компоненты, описанные в следующей таблице, в центральная зона дизайна. Расположите компоненты, как показано на предыдущем изображении.

Составная часть	Имя в коде	Недвижимость не по умолчанию	Ярлык, связанный с Составная часть
Топоры	AxesFreq	Непригодный	частота
Топоры	AxesTime	Непригодный	Время
Числовое поле редактирования	NumericEditFieldf1	Максимум: 500	f1
Числовое поле редактирования	NumericEditFieldf2	Максимум: 500	f2
Редактировать поле	EditFieldTime	Непригодный	T

Составная часть	Имя в коде	Недвижимость не по умолчанию	Ярлык, связанный с Составная часть
кнопка	кнопка	Текст: Сюжет	Непригодный

- 2
- Щелкните правой кнопкой мыши на **Plot** кнопку и выберите **Callbacks > Edit ButtonPushedFcn** .
- 3
- В диалоговом окне Добавить функцию обратного вызова нажмите **ОК** . (Примите обратный вызов по умолчанию имя функции.)
- Дизайнер приложений открывает представление кода.
- 4
- Получите входные значения, выполните расчеты и нанесите данные на соответствующие оси.

Начиная с строки следующей функции ButtonButtonPush (приложение) добавить:

```
% Получить пользовательский ввод
f1 = app.NumericEditFieldf1.Value;
f2 = app.NumericEditFieldf2.Value;
t = eval (app.EditFieldTime.Value);

% Рассчитать данные
x = sin (2 * pi * f1 * t) + sin (2 * pi * f2 * t);
y = фут (x, 512);
m = y. * cos (y) / 512;
f = 1000 * (0: 256) / 512;

% Создание частотного графика по соответствующим осям
участок (app.AxesFreq, e, m (1: 257))
app.AxesFreq.XMinorTick = 'on';
сетка на
```

```
% Создать график времени в правильных осях
участок (app.AxesTime, t, x)
app.AxesTime.XMinorTick = 'on';
сетка на
```

5 Сохраните и запустите приложение.

На вкладке « **Редактор** » нажмите « **Запустить приложение** » .

Сохраните приложение при появлении запроса.

Протестируйте приложение, попробовав различные элементы управления приложения в запущенном приложении. Заметить, что:

- Числовые поля редактирования, f1 и f2, отображают сообщение об ошибке, если вы вводите нечисловое значение или значение, которое превышает установленное вами максимальное значение.

3-24

- Поле редактирования для вектора времени не мешает вам ввести неверный вектор.  
Для обработки недопустимых значений вектора времени определите функцию обратного вызова ValueChangedFcn для компонент EditFieldTime, а затем закодируйте его, чтобы определить, если записи пользователя действительны.

### Приложение, отображающее два графика в одной оси

В этом примере показано, как создать приложение, позволяющее пользователю управлять отображаемыми данными. в осях с помощью кнопок. Кнопки создают разные графики на одних и тех же осях.

После того, как пользователь нажимает кнопку, он отключается. Поскольку данные, наносимые на график, не изменить, нет никаких причин для пользователя, чтобы нажать каждую кнопку более одного раза.

Чтобы создать приложение:

3-25

- 1
- В App Designer перетащите оси и две кнопки в центральную область дизайна. организовать их, как показано на предыдущем изображении и как описано в этой таблице.

Составная часть	Текст	Имя в коде
кнопка	Участок 1 (линия x)	Plot1
кнопка	Участок 2 (линия o)	Plot2
- 2
- Нажмите **Вид кода** .
- 3
- Добавьте свойство приложения для хранения данных для вашего приложения.

В браузере кода нажмите **Добавить свойство** и введите X в открывшемся поле.

App Designer добавляет имя вместе с приложением. префикс.
- 4
- Укажите значение для оси X в функции запуска.

На вкладке « **Редактор** » нажмите « **Просмотреть функцию запуска** » . Добавьте этот код в автозагрузку функция:

```
app.X = пробел (-2 * пи, 2 * пи);
```
- 5
- Создайте обратный вызов для кнопки Plot1.

В браузере кода щелкните правой кнопкой мыши **app.Plot1** и выберите **Callbacks > Edit. ButtonPressedFcn** .
- 6
- В диалоговом окне Добавить функцию обратного вызова нажмите **ОК** . (Примите обратный вызов по умолчанию имя функции.)
- 7
- В коде, начинающемся на строке после функции Plot1ButtonPush (app), добавьте следующий код:

```
y = грех (приложение X);  
сюжет (app.Axes, app.X, y, 'x' )  
удерживать (app.Axes, 'on' );  
app.Plot1.Enabled = false;
```

Последний входной аргумент для функции plot указывает, что график будет представлены серией x символов.

Оператор Hold гарантирует, что дополнительные участки не будут перезаписывать существующие участки в указанных осях.
- 8
- Создайте обратный вызов для кнопки Plot2.

3-26

- 
- В коде браузера, щелкните правой кнопкой мыши **app.Plot2** и выберите **Callbacks > Edit ButtonPressedFcn** .
- 9
- В диалоговом окне Добавить функцию обратного вызова нажмите **ОК** . (Примите обратный вызов по умолчанию имя функции.)



10 В коде, начинающемся на строке после функции Plot2ButtonPush (app), добавьте следующий код:

```
y = cos (app.X);
сюжет (app.Axes, app.X, y, 'o' )
удерживать (app.Axes, 'on' );
app.Plot2.Enabled = false;
```

11 Сохраните и запустите приложение.

На вкладке « **Редактор** » нажмите « **Запустить приложение** » .

Сохраните приложение при появлении запроса.

Протестируйте приложение, испробовав различные элементы управления в запущенном приложении.

### Приложение, отображающее значение ползунка по мере его изменения

В этом примере показано, как создать приложение, отображающее значение ползунка в поле редактирования.

Поле редактирования постоянно обновляется, чтобы отразить текущее значение ползунка. (Другой пример, [«Создание и запуск простого приложения» на стр. 1-3](#), обновляет поле редактирования только тогда, когда приложение пользователь отпускает кнопку мыши с ползунка.)

3-27

Чтобы создать приложение:

- 1 В App Designer перетащите **ползунок** и **поле «Числовое редактирование»** в центральный дизайн-область, расположив их, как показано на предыдущем изображении.
- 2 Щелкните правой кнопкой мыши на ползунок и выберите **Callbacks > ValueChangingFcn** .
- 3 В диалоговом окне Добавить функцию обратного вызова нажмите **ОК** . (Примите обратный вызов по умолчанию имя функции.)
- 4 В коде, начинающемся на строке после функции

- SliderValueChanging (приложение, событие) добавить этот код:  
app.NumericEditField.Value = event.Value;
- 5 Сохраните и запустите приложение.

На вкладке « Редактор » нажмите « Запустить приложение » .

Сохраните приложение при появлении запроса.

Протестируйте приложение, выполнив одно из следующих действий:

- Перетащите ползунок на новое значение.

3-28

- При этом числовое поле редактирования постоянно обновляется для отображения текущего значение слайдера.
- Нажмите ползунок.

Числовое поле редактирования обновляется для отображения нового значения ползунка.

Приложение для изменения цвета лампы на основе выбора группы кнопок

В этом примере показано, как создать приложение, которое меняет цвет лампы на основе какую радиокнопку выбирает пользователь.

Чтобы создать приложение:

- 1 В конструкторе приложений перетащите группу **радиокнопок** , **лампу** и **метку** в центральная зона дизайна. Расположите их, как показано на предыдущем изображении и описано в этот стол.

Составная часть	Название или текст	Имя в коде
<b>ButtonGroup</b>	Показатель	ButtonGroup (по умолчанию)
<b>Переключатель</b>	Стоп	StopRadioButton
<b>Переключатель</b>	предосторожность	CautionRadioButton

3-29

Составная часть	Название или текст	Имя в коде
<b>Переключатель</b>	Идти	GoRadioButton
<b>Лампа</b>	Непригодный	Лампа (по умолчанию)
<b>этикетка</b>	Статус	Метка (по умолчанию)

- 2
- Выберите радио кнопку **Go** .
- 3
- На вкладке « **Свойства** » установите флажок « **Выбрано** » .
- 4
- Нажмите **Вид кода** .
- 5
- В коде браузера, щелкните правой кнопкой мыши **app.ButtonGroup** , а затем выберите **Callbacks > SelectionChangeFcn** .
- 6
- В диалоговом окне Добавить функцию обратного вызова нажмите **ОК** . (Примите обратный вызов по умолчанию имя функции.)
- 7
- Добавьте код, чтобы определить, какой переключатель выбран, и установите цвет лампы. соответственно.

В коде, начинающемся на строке после функции  
ButtonGroupSelectionChange (приложение, событие), добавьте:

```
переключатель app.ButtonGroup.SelectedObject
    чехол приложение. GoRadioButton
        app.Lamp.Color = 'green' ;
    case app.CautionRadioButton
        app.Lamp.Color = 'yellow' ;
    case app.StopRadioButton
        app.Lamp.Color = 'red' ;
конец
```

- 8
- Сохраните и запустите приложение.
- На вкладке « **Редактор** » нажмите « **Запустить приложение** » .

Сохраните приложение при появлении запроса.

Протестируйте приложение, выбрав каждый переключатель.

Приложение, которое изменяет размер компонента при изменении размера окна

В этом примере показано, как кодировать функцию обратного вызова для окна приложения, чтобы при изменении размера пользователя окно, ползунок внутри окна изменяет размеры соответственно, оставаясь в центре окно.

Чтобы создать приложение:

- 1
- В App Designer перетащите **ползунок** в середину центральной области дизайна.
- 2
- Нажмите **Вид кода** .
- 3
- В коде браузера, щелкните правой кнопкой мыши **app.AppWindow** , а затем выберите **Callbacks > Редактировать SizeChangedFcn** .

- 4 В диалоговом окне Добавить функцию обратного вызова нажмите **ОК** . (Примите обратный вызов по умолчанию имя функции.)
- 5 На месте курсора в представлении кода, начиная со строки после функции  
AppWindowSizeChange (app) добавить этот код.

```

windowpos = app.AppWindow.Position;
slidersize = app.Slider.Size;
% Вычтите высоту слайдера из
% высоты окна и делим на 2.
app.Slider.Location = [30, (windowpos (4) -6) / 2];
% Вычтите 60 из ширины окна.
app.Slider.Size = [windowpos (3) -60, slidersize (2)];

```

Код получает ширину и высоту окна приложения после операции изменения размера. Код обновляет расположение и размер ползунка, чтобы держать его в центре окна приложения, 30 пикселей с левой и правой сторон окна.

3-31

---

## Страница 52

### 3 Программирование приложений

- 6 Сохраните и запустите приложение.  
На вкладке « **Редактор** » нажмите « **Запустить приложение** » .  
Сохраните приложение при появлении запроса.  
Протестируйте приложение, изменив размер окна приложения.

3-32

---

## Страница 53

# Горячие клавиши

- « [Ярлыки выбора компонентов](#)» на стр. [4-2](#)
- « [Ярлыки перемещения компонентов](#)» на стр. [4-3](#)
- « [Ярлыки изменения размера компонентов](#)» на стр. [4-4](#)
- « [Копирование, вставка, дублирование и удаление ярлыков компонентов](#)» на стр. [4-5](#).
- « [Ярлыки выравнивания компонентов](#)» на стр. [4-6](#)
- « [Сочетания клавиш для восстановления, отмены и отмены операции](#)» на стр. [4-7](#).
- « [Запуск, сохранение и открытие ярлыков приложений](#)» на стр. [4-8](#).

---

## Страница 54

4 сочетания клавиш

### Ярлыки выбора компонентов

действие	Ключ или Ключи
Переместить выделение на следующий компонент, контейнер или окно приложения в дизайне последовательность навигации по клавише Tab.	<b>табуляция</b>
Переместить выбор к предыдущему окну компонента, контейнера или приложения в последовательность навигации по вкладке области дизайна.	<b>Shift + Tab</b>
Выбрать все выбираемые компоненты, кроме те, которые являются детьми другого составная часть.	<b>Ctrl + A,</b>
Отменить выбор ранее выбранного компонента или повторно выбрать невыбранный компонент (выбор).	Нажмите <b>Shift + клик</b> .

4-2

Страница 55

Ярлыки перемещения компонентов

## Ярлыки перемещения компонентов

действие	Ключ или Ключи
Переместить выбранные компоненты вниз на 1 пиксель	<b>Стрелка вниз</b>
Переместить выбранные компоненты влево на 1 пиксель	<b>Стрелка влево</b>
Переместить выбранные компоненты вправо на 1 пиксель	<b>Правая стрелка</b>
Переместить выбранные компоненты вверх на 1 пиксель	<b>Стрелка вверх</b>
Переместить выбранные компоненты вниз на 10 пикселей.	<b>Shift + стрелка вниз</b>
Переместить выбранные компоненты влево на 10 пикселей.	<b>Shift + Стрелка влево</b>
Переместить выбранные компоненты вправо на 10 пикселей.	<b>Shift + стрелка вправо</b>
Переместить выбранные компоненты вверх на 10 пикселей.	<b>Shift + стрелка вверх</b>
Отмените текущую операцию.	<b>Побег</b>

Страница 56

4 сочетания клавиш

Ярлыки изменения размера компонента

действие	ключ
Изменение размера компонента при сохранении аспекта соотношения.	Нажмите и удерживайте клавишу <b>Shift</b> до Вы начинаете перетаскивать компонент расширителя.
Отмените текущую операцию изменения размера.	<b>Побег</b>

4-4

Страница 57

Копирование компонентов, вставка, дублирование и удаление ярлыков

Копирование компонентов, вставка, дублирование и удаление ярлыков

действие	Ключ или Ключи
Скопируйте выбранные компоненты в буфер обмена.	<b>Ctrl + C ,</b>
Дублировать выбранный компонент (без копирование в буфер обмена).	<b>Ctrl + D,</b>
Вставьте компоненты из буфера обмена в дизайн области.	<b>Ctrl + V,</b>
Вырезать выбранные компоненты (копируя их в буфер обмена и удаление их из область дизайна).	<b>Ctrl + X</b>

Удалить выбранные компоненты из  
дизайн области.

**Backspace** или **Удалить**

4-5

4 сочетания клавиш

Ярлыки выравнивания компонентов

действие	Ключи
Выровняйте выбранные компоненты слева кромки.	<b>CTRL + 1</b>
Совместите выбранные компоненты на их горизонтальные центры.	<b>CTRL + 2</b>
Совместите выбранные компоненты справа кромки.	<b>CTRL + 3</b>
Выровнять выбранные компоненты по их верху кромки.	<b>CTRL + 4</b>
Выровнять выбранные компоненты по вертикали средний.	<b>CTRL + 5</b>
Выровнять выбранные компоненты по их низу кромки.	<b>CTRL + 6</b>



Восстановить, отменить и отменить ярлыки операций

действие	Ключ или Ключи
Вернуть отмененную модификацию области дизайна, вернуть его в измененное состояние.	<b>Ctrl + Y</b> или <b>Ctrl + Shift + Z</b>
Вернуть отмененную модификацию редактора кода, вернуть его в измененное состояние.	<b>Ctrl + Y</b>
Отменить область дизайна или редактор кода модификация, возвращая ее к предыдущему государству.	<b>Ctrl + Z</b>
Отмена текущего изменения размера или перемещения операция.	<b>Побег</b>

Запускать, сохранять и открывать ярлыки приложений

действие	Ключ или Ключи
Сохраните активное приложение.	<b>Ctrl + S</b>
Запустите активное приложение.	<b>F5</b>
Откройте ранее сохраненное приложение.	<b>Ctrl + O</b>

4-8

---

Страница 61

5

## Ссылка на недвижимость

[Свойства окна приложения](#)  
[Свойства осей](#)  
[Свойства кнопки](#)  
[Свойства группы кнопок](#)  
[Флажок Свойства](#)  
[Свойства дискретной ручки](#)  
[Drop Down Properties](#)  
[Изменить свойства поля](#)  
[Редактируемые раскрывающиеся свойства](#)  
[Калибровка недвижимости](#)  
[Knob Properties](#)  
[Свойства ярлыка](#)  
[Свойства лампы](#)  
[Свойства линейного датчика](#)  
[Свойства списка](#)  
[Свойства децимально градусов](#)  
[Числовые свойства поля редактирования](#)  
[Свойства панели](#)  
[Свойства радиокнопок](#)  
[Свойства клавишного переключателя](#)  
[Свойства полукруглого датчика](#)  
[Свойства слайдера](#)  
[Свойства кнопки состояния](#)  
[Переключить свойства](#)  
[Свойства текстовой области](#)

[Свойства кнопки переключения](#)  
[Свойства тумблера](#)

---

## Страница 62

5 Ссылка на недвижимость

# Свойства окна приложения

Управление окном и поведением окна приложения

Окна приложения содержат приложение, которое вы создаете с помощью App Designer. Окно приложения — это рисунок специально для дизайнера приложений. Свойства фигуры, которые поддерживаются и относящиеся к рабочему процессу дизайнера приложений, документированы в этом разделе документации. Свойства контролируют внешний вид и поведение конкретного экземпляра окна приложения. Чтобы изменить аспекты окна приложения, измените значения свойств. Используйте точку запятой для обозначения конкретного объекта и свойства:

```
window = appwindow;  
name = window.Name;  
window.Name = 'Мои результаты';
```

**Дети** - Объекты, для которых Окно Приложения является родителем  
пустой массив GraphicsPlaceholder (по умолчанию) | вектор объектов

Объекты, для которых App Window является родительским, указываются как массив объектов.

Чтобы изменить расположение объектов на дисплее, измените их порядок в векторе.

Пример: [o1, o2, o3, o4]

где каждый *o* является объектом.

**Имя** - заголовок окна приложения  
«Окно приложения» (по умолчанию) | строка

5-2

---

## Страница 63

Свойства окна приложения

Заголовок окна приложения, указанный в виде строки.

Пример: «Результаты»

**Родитель** - родитель окна приложения  
корневой объект

Родитель окна приложения, возвращается как корневой объект.

**Положение** - размер и расположение окна приложения  
[Слева, снизу, ширина, высота]

Размер и расположение окна приложения на экране (исключая строку заголовка, строку меню, панели инструментов и внешние края), определяется как [left, bottom, width, height].

Значения слева и снизу определяют расстояние от нижнего левого угла экрана.  
в нижний левый угол окна приложения. Значения ширины и высоты определяют  
Размеры окна. Левый и нижний элементы могут быть отрицательными в системах  
которые имеют более одного монитора. Все измерения в пикселях.

Пример: [230 250 570 510]

**SizeChangedFcn** - функция обратного вызова, которая выполняется при изменении размера окна приложения  
ручка функции | массив ячеек | строка

Функция обратного вызова, которая выполняется при изменении размера окна приложения, указанного как  
дескриптор функции, массив ячеек, содержащий дескриптор функции и дополнительные аргументы, или  
строка. MATLAB выполняет эту процедуру обратного вызова, когда происходит любое из следующего:

- Пользователь вручную изменяет размер окна приложения.
- Код изменяет размеры окна приложения.

Во время выполнения функции обратного вызова размер дескриптора окна приложения изменяется  
доступен только через корневое свойство [CallbackObject](#), которое можно запросить с помощью  
gcbo.

Если вы измените Позицию окна приложения из обратного вызова SizeChangedFcn,  
SizeChangedFcn больше не вызывается.

Пример: @myfun

Пример: {@ myfun, x}

Пример: {'myfun', x}

5-3

## Страница 64

5 Ссылка на недвижимость

**Единицы** - Единицы измерения

«пикселей» (по умолчанию) | 'дюймов' | сантиметры | «нормализованный» | «очки» |  
'персонажи'

Единицы измерения, указанные как «пиксели», «дюймы», «сантиметры»,  
«нормализованные», «точки» или «символы».

Примечание. Независимо от указанного вами значения свойства Units, MATLAB всегда использует  
пиксельные единицы.

**Видимый** - видимость окна приложения

'on' (по умолчанию) | «Выключено»

Видимость окна приложения, указанная как «включено» или «выключено». Видимое свойство определяет  
отображается ли окно приложения на экране. Если видимое свойство приложения  
окно выключено, все окно приложения скрыто, но вы все равно можете указать и  
получить доступ к его свойствам.

Изменение размера скрытого окна приложения запускает обратный вызов SizeChangedFcn, если и  
когда окно приложения становится видимым.

Примечание Изменения Visible свойства окна приложения никак *не* изменят значения  
Видимые свойства его дочерних компонентов, даже если скрытие окна приложения вызывает  
компоненты должны быть скрыты.

## Свойства осей

Управление осями внешнего вида и поведения

Свойства осей управляют внешним видом и поведением объекта осей. Путем изменения значения свойств, вы можете изменить некоторые аспекты осей. Используйте точечную запись для обозначения конкретный объект и свойство:

```
window = appwindow;  
ax = axes ('Parent', окно);  
цвет = топор. цвет;  
ax.Color = 'blue';
```

- ActivePositionProperty** - свойство Position для сохранения константы во время операции изменения размера 'внешнее положение' (по умолчанию) | 'позиция'
- Свойство Position для сохранения константы во время операции изменения размера, указанной как один из этих ценности:
- 'externalposition' - содержит свойство OuterPosition постоянным. Если вы измените размер Окно приложения, макет окна регулируется в соответствии с xlabel, ylabel и заголовки строк.
  - 'position' - удерживать свойство Position постоянным. Если вы измените размер окна приложения, расположение окна не регулируется для размещения строк xlabel, ylabel и title.

Окно приложения может изменять размер либо в интерактивном режиме, либо во время печати или экспорта операция.

**BeingDeleted** - Статус удаления осей

'выкл' (по умолчанию) | 'на'

Состояние удаления осей, возвращенное как «включено» или «выключено». MATLAB устанавливает бытие свойство on, когда функция удаления осей начинает выполнение (см.

Свойство DeleteFcn). Свойство BeingDeleted остается включенным до тех пор, пока оси не дольше существует.

Проверьте значение свойства BeingDeleted, чтобы убедиться, что оси не будут удаляться перед запросом или изменением свойств осей.

**Коробка** - контур коробки с осями

'выкл' (по умолчанию) | 'на'

Контур поля Оси, указанный как одно из следующих значений:

- «выкл» - контур окна не отображается вокруг осей.
- «вкл» - отображает контур рамки вокруг осей.

Свойства XColor и YColor управляют цветом контура.

**Дети** - объект детей топоров

пустой массив GraphicsPlaceholder (по умолчанию) | массив графических объектов

Дочерние объекты оси, возвращаемые как массив графических объектов. Вы не можете добавить или удалить детей, используя свойство осей Children.

Чтобы добавить дочерний элемент в этот список, задайте свойство Parent дочернего графического объекта для осей. объект.

**CLim** - сопоставление цветов для объектов с использованием карты цветов

[0 1] (по умолчанию) | [cmin smax]

Отображение цветов для объектов на осях, использующих карту цветов, указанную в виде двухэлементного вектор вида [cmin smax].

Свойство CLim определяет, как MATLAB отображает значения данных в цвета в карта цветов, где:

- cmin указывает значение данных, которое отображается на первый цвет в карте цветов.
- smax указывает значение данных, которое отображается на последний цвет в карте цветов.

Оси линейно интерполируют значения данных между cmin и smax по всей карте цветов.

Значения за пределами этого диапазона используют либо первый, либо последний цвет карты цветов, в зависимости от того, что ближе всего.

5-6

Для получения информации об изменении карты цветов см. Функцию `colormap`. Для информации о цветовом отображении см. функцию `axis`.

Установка этого свойства устанавливает для свойства `CLimMode` значение «manual».

**CLimMode** - режим выбора для **CLim**

'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим выбора для свойства `CLim`, указанного в качестве одного из следующих значений:

- 'auto' - установить свойство `CLim` так, чтобы оно охватывало пределы `CData` графических объектов. отображается в осях.
- «ручной» - используйте значения `CLim`, указанные вручную. Чтобы указать значения, установите `CLim` невидимостью. Оси не изменяют значение свойства `CLim`, когда пределы осей у детей меняются.

**Color** - цвет осей задних плоскостей

[1 1 1] (по умолчанию) | RGB триплет | цветная строка | 'никто'

Цвет задних плоскостей осей, заданный как триплет RGB, цветная строка или «нет». Если Вы устанавливаете цвет 'none', тогда оси невидимы, и цвет окна приложения показывает через.

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне [0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«Y»	[1 1 0]
«Малиновый»	«M»	[1 0 1]
«Голубой»	«C»	[0 1 1]
«Красный»	'r'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«Б»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный	«К»	[0 0 0]
Пример: [0 0 1]		

5-7

Пример: 'b'

Пример: «синий»

**ColorOrder** - Цвета для многострочных графиков  
семь предопределенных цветов (по умолчанию) | трехколонная матрица триплетов RGB

Цвета для многострочных графиков, заданные в виде трехколонной матрицы триплетов RGB, где каждый строка определяет один цвет в цветовом порядке. Функции, которые создают линейные графики, пролистывают цвета, определенные свойством ColorOrder для окраски каждой построенной линии.

Порядок цветов по умолчанию имеет семь цветов.

Порядок цветов по умолчанию	Связанные Тройки RGB
	[0 0,4470 0,7410
	0,8500 0,3250 0,0980
	0,9290 0,6940 0,1250
	0,4940 0,1840 0,5560
	0,4660 0,6740 0,1880
	0,3010 0,7450 0,9330
	0,6350 0,0780 0,1840]

Функции, которые создают линейные графики, циклически изменяют цвета для каждой линии. Трюм Состояние для осей влияет на используемые цвета:

- Если состояние удержания для осей выключено (когда свойство NextPlot установлено в «replace», затем высокоуровневые функции построения графиков, такие как plot, сбрасывают порядок цветов в значения по умолчанию цвета перед нанесением. Если вы хотите указать новые цвета для порядка цвета и не хотите, чтобы высокоуровневые функции построения графиков сбрасывали их, затем устанавливайте NextPlot свойство «заменить детям». Кроме того, вы можете указать новый по умолчанию значение для свойства ColorOrder в корне с помощью функции set. Например, `set (groot, 'defaultAxesColorOrder', [0 1 0; 0 0 1])`.
- Если состояние удержания от осей включено (когда для свойства NextPlot установлено значение «добавить»), затем функции построения графика перебирают цвета, начиная с места в цвете порядок, где последний участок закончился.

Типы данных: один | двойной

**CreateFcn** - создание обратного вызова  
" (по умолчанию) | ручка функции | массив ячеек | строка

Обратный вызов создания, указанный как одно из следующих значений:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимой командой или функцией MATLAB, которая оценивается в базе рабочее пространство (не рекомендуется)

Используйте это свойство для выполнения кода при создании осей. Установка `CreateFcn` Свойство на существующие оси не влияет. Вы должны определить значение по умолчанию для этого или определите это свойство с помощью пары `Имя, Значение` при создании осей. MATLAB выполняет обратный вызов после создания осей и установки всех его свойств.

Если вы укажете этот обратный вызов, используя дескриптор функции, MATLAB передает два аргумента функция обратного вызова при выполнении обратного вызова:

- Объект оси - Вы можете получить доступ к свойствам объекта оси из функция обратного вызова. Вы также можете получить доступ к объекту осей через `CallbackObject` свойство корня, к которому можно обратиться с помощью функции `getco`.
- Данные события - этот аргумент пуст для этого свойства. Замените это тильдой символ (~) в определении функции, чтобы указать, что этот аргумент не используется.

Для получения дополнительной информации о том, как использовать дескрипторы функций для определения функций обратного вызова, см. Тема «Определение обратного вызова» в документации MATLAB.

Пример: `@myCallback`

Пример: `{@ myCallback, arg3}`

**DataAspectRatio** - Относительная длина блоков данных вдоль каждой оси `[1 1 1]` (по умолчанию) | трехэлементный вектор вида `[dx dy n]`

Относительная длина блоков данных вдоль каждой оси, указанная в виде трехэлементного вектора форма `[dx dy n]`. Первые два векторных элемента определяют относительные данные `x` и `y` масштабные факторы. Третий векторный элемент игнорируется, но должен быть включен. Например, указав это свойство как `[1 2 1]` устанавливает длину одной единицы данных в `x` -направлении быть такой же длины, как и две единицы данных в `y` -направлении.

Свойство `DataAspectRatio` взаимодействует с `PlotBoxAspectRatio`, `XLimMode`, и свойства `YLimMode` для управления тем, как MATLAB масштабирует ось `X` и `Y`.

Установка значения для `DataAspectRatio` отключает «растягивать, чтобы заполнить окно приложения» поведение формы, если `DataAspectRatioMode` и `PlotBoxAspectRatioMode` оба установлены в 'авто'.

5-9

Если для соответствующего свойства `mode` установлено значение «auto», то MATLAB выбирает значения отношения. Если вы установите это свойство, то MATLAB установит режим на «ручной».

Пример: `[1 1 1]`

Типы данных: один | двойной | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64`

**DataAspectRatioMode** - режим выбора для **DataAspectRatio** 'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим выбора для `DataAspectRatio` указывается в качестве одного из следующих значений:

- «Авто» - используйте значения, которые наилучшим образом используют область, предоставленную окном приложения.
- «ручной» - используйте значения, указанные вручную. Чтобы указать значения, установите `DataAspectRatioMode` to 'manual'. Изменение `DataAspectRatioMode` на «ручной» отключает поведение «растягивать, чтобы заполнить форму окна приложения», если `DataAspectRatioMode` и значения свойств `PlotBoxAspectRatioMode` установлены в «auto».



В этой таблице описывается взаимодействие между свойствами при отключении растяжения заполнить поведение.

XLimitMode и YLimitMode	DataAspectRatioMode	PlotBoxAspectRatioMode Behavior
'авто'	'авто'	'авто'Пределы, выбранные для охвата диапазон данных во всех размеры.
'авто'	'авто'	'руководство по эксплуатации'Выбранные ограничения диапазон данных диапазона во всех измерениях. MATLAB модифицирует DataAspectRatio к достичь запрошенного PlotBoxAspectRatio в пределах программное обеспечение выбрано.
'авто'	'руководство по эксплуатации'	'авто'Выбранные ограничения диапазон данных диапазона во всех измерениях. MATLAB модифицирует

5-10

Страница 71

Свойства осей

XLimitMode и YLimitMode	DataAspectRatioMode	PlotBoxAspectRatioMode Behavior
		PlotBoxAspectRatio к достичь запрошенного DataAspectRatio в пределах программное обеспечение выбрано.
'авто'	'руководство по эксплуатации'	'руководство по эксплуатации'Выбранные ограничения полностью подходит и центрировать сюжет в пределах запрошенного PlotBoxAspectRatio учитывая запрошенный DataAspectRatio (это может производить пустой пространство около 2 из 3 размеры).
'руководство по эксплуатации'	'авто'	'авто'MATLAB отличием ограничивает и изменяет DataAspectRatio и PlotBoxAspectRatio as необходимо.
'руководство по эксплуатации'	'авто'	'руководство по эксплуатации'MATLAB отличием пределы и PlotBoxAspectRatio и модифицирует DataAspectRatio as необходимо.
'руководство по эксплуатации'	'руководство по эксплуатации'	'авто'MATLAB соблюдает ограничения и DataAspectRatio и модифицирует PlotBoxAspectRatio as необходимо.

DeleteFcn - удаление обратного вызова  
" (по умолчанию) | ручка функции | массив ячеек | строка  
Обратный вызов удаления, указанный как одно из следующих значений:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимой командой или функцией MATLAB, которая оценивается в базе рабочее пространство (не рекомендуется)

Используйте это свойство для выполнения кода при удалении осей. MATLAB выполняет обратный вызов до уничтожения осей, чтобы обратный вызов мог получить доступ к значениям его свойств.

Если вы укажете этот обратный вызов, используя дескриптор функции, MATLAB передает два аргумента функция обратного вызова при выполнении обратного вызова:

- Объект оси - Вы можете получить доступ к свойствам объекта оси из функция обратного вызова. Вы также можете получить доступ к объекту осей через CallbackObject свойство корня, к которому можно обратиться с помощью функции gcbo.
- Данные события - этот аргумент пуст для этого свойства. Замените это тильдой (~) в определении функции, чтобы указать, что этот аргумент не используется.

Для получения дополнительной информации о том, как использовать дескрипторы функций для определения функций обратного вызова, см. Тема «Определение обратного вызова» в документации MATLAB.

Пример: @myCallback

Пример: {@ myCallback, arg3}

**FontAngle - Наклон персонажа**  
'нормальный' (по умолчанию) | «Наклонным»

Наклон символов, указанный как «обычный» или «курсив». Установка этого свойства курсивом выбирает наклонную версию шрифта, если она доступна в системе пользователя приложения. Не все шрифты имеют оба стиля шрифта, поэтому курсивный шрифт может выглядеть так же, как обычный шрифт.

**FontSize - Размер шрифта**  
10 (по умолчанию) | скалярное числовое значение

Размер шрифта, указанный как скалярное числовое значение. Свойство FontSize определяет размер текста, используемого для меток оси и заголовка. Свойство FontUnits определяет Единицы, используемые для интерпретации размера шрифта.

Пример: 12

**FontUnits - Единицы размера шрифта**  
'points' (по умолчанию) | 'дюймов' | сантиметры | «персонажи» | «Нормализованы» | «пикселей»

Единицы размера шрифта, указанные как одно из значений в следующей таблице.

Единицы	Описание
'точки'	Точки. Одна точка равна 1/72 дюйма.
«» дюймов	дюймов
«» сантиметров	сантиметров

'персонажи'	На основе размера символов в системный шрифт по умолчанию. Ширина одного символьная единица - ширина буквы  Икс. Высота одной символьной единицы расстояние между базовыми линиями двух линий текста.
«Нормализованы»	Интерпретируется как доля высоты осей. Если вы измените размеры осей, MATLAB изменит размер шрифта соответственно. Например, если Размер шрифта равен 0,1 в нормализованных единицах, тогда текст составляет 1/10 от высоты осей.
«пикселей»	Пиксели. Размер пикселя зависит от экрана разрешающая способность.

Если вы устанавливаете FontSize и FontUnits в одном вызове функции, вы должны установить Сначала свойство FontUnits, чтобы оси правильно интерпретировали указанный FontSize.

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'  
Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**GridColor** - цвет линий сетки  
[0,15 0,15 0,15] (по умолчанию) | RGB триплет | цветная строка | 'никто'

Цвет линий сетки, указанный как триплет RGB, цветная строка или «нет».

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне [0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«У»	[1 1 0]
«Малиновый»	«М»	[1 0 1]
«Голубой»	«С»	[0 1 1]
«Красный»	'р'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«Б»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный	«К»	[0 0 0]

Чтобы установить цвета для контура поля осей, используйте свойства XColor и YColor.

Пример: [0 0 1]

Пример: 'b'

Пример: «синий»

**GridColorMode** - режим выделения для **GridColor**  
'auto' (по умолчанию) | 'руководство по эксплуатации'  
Режим выбора для свойства GridColor, заданный как «автоматический» или «ручной». Цвет на основе значений свойств GridColorMode, XColorMode и YColorMode.

В этой таблице перечислены цвета линий сетки для различных комбинаций цветовых режимов.

Свойство GridColorMode	Свойство XColorMode	Цвет линии сетки оси x
------------------------	---------------------	------------------------

'авто'	'авто'	Определяется GridColor свойство
'авто'	'руководство по эксплуатации'	Определить по XGridColorproperty

5-14

Страница 75

Свойства осей

Свойство GridColorMode	Свойство XColorMode	Цвет линии сетки оси x
'руководство по эксплуатации'	'авто'	Определяется GridColor свойство
'руководство по эксплуатации'	'руководство по эксплуатации'	Определяется GridColor свойство
Свойство GridColorMode	Свойство YColorMode	Цвет линии сетки оси Y
'авто'	'авто'	Определяется GridColor свойство
'авто'	'руководство по эксплуатации'	Определить по YGridColorproperty
'руководство по эксплуатации'	'авто'	Определяется GridColor свойство
'руководство по эксплуатации'	'руководство по эксплуатации'	Определяется GridColor свойство

**LineStyleOrder** - стили линий и маркеры для многострочных графиков  
'-' сплошная линия (по умолчанию) | массив ячеек спецификаторов  
Стили линий и маркеры для многострочных графиков, заданные как массив ячеек любого числа и Сочетание спецификаторов в этой таблице.

Тендерный	Стиль линии
'-'(по умолчанию)	Сплошная линия
'+'	Маркеры со знаком плюс
«О»	Круговые маркеры
'*'	Звездные маркеры
"	Точечные маркеры
'Икс'	Перекрестные маркеры
'S'	Квадратные маркеры
'D'	Алмазные маркеры
'^'	Направленные вверх треугольные маркеры
'v'	Направленные вниз треугольные маркеры
'>'	Правильные треугольные маркеры

5-15

Страница 76

5 Ссылка на недвижимость

Тендерный	Стиль линии
'<'	Левый указатель треугольника
'п'	Маркеры пятиконечной звезды (пентаграммы)
'час'	Маркеры шестиконечной звезды (гексаграммы)

Объедините строки, чтобы указать строки с маркерами, такими как '- \*'. Сюжет циклически проходит через стили линий только после использования всех цветов, содержащихся в свойстве ColorOrder. Если удержание состояние для осей выключено (когда свойство NextPlot для осей установлено в «замена»), затем высокоуровневые функции построения графиков, такие как plot, сбрасывают порядок стилей линий по умолчанию стили линий перед построением.

Если вы хотите указать набор стилей линий, который отличается от стандартного и не хотите, чтобы высокоуровневые функции построения графиков сбросили свойство LineStyleOrder, затем установите Следующий сюжет для «замены детей». Кроме того, вы можете указать новое значение по умолчанию для свойства LineStyleOrder в корне, используя функцию set. Например, набор (Гроот, 'defaultAxesLineStyleOrder', {'- \*', '!', 'o'}).

Пример: {'- \*', '!', 'o'}

**LineWidth** - ширина контуров осей, отметок и линий сетки  
0,5 (по умолчанию) | скалярное значение

Ширина контуров осей, отметок и линий сетки, указанная в виде скалярного значения в точке единицы. Одна точка равна 1/72 дюйма. Значение по умолчанию составляет 0,5 балла.

Пример: 1,5

**MinorGridColor** - цвет второстепенных линий сетки  
[0,1 0,1 0,1] (по умолчанию) | RGB триплет | цветная цепочка

Цвет второстепенных линий сетки, указанный как триплет RGB, цветная строка или «нет».

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне [0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«Y»	[1 1 0]
«Малиновый»	«M»	[1 0 1]

Длинное имя	Короткое имя	RGB Триплет
«Голубой»	«C»	[0 1 1]
«Красный»	'r'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«Б»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный	«К»	[0 0 0]

Пример: [0 0 1]  
Пример: 'b'  
Пример: «синий»

**NextPlot** - свойства для сброса при добавлении нового графика  
«заменить» (по умолчанию) | «добавить» | 'Replacechildren'

Свойства для сброса при добавлении нового графика к осям, указанные в качестве одного из следующих значений:

- «заменить» - сбросить все свойства осей, кроме позиции и единиц, по умолчанию Значения и удалите все дочерние оси перед отображением нового графика. Использование этого значения аналогично использованию сброса кла.
- «добавить» - использовать существующие оси, чтобы добавить больше графиков к осям и не сбрасывать свойства.
- replacechildren - удаляет все дочерние объекты перед добавлением новых графиков, но не сбросить свойства осей. Это похоже на использование кла.

Функция newplot упрощает использование свойства NextPlot и полезна для написание пользовательских графических функций.

**OuterPosition** - размер и расположение осей, включая метки, заголовок и поля  
`[0 0 1 1]` (по умолчанию) | `[` высота левого нижнего края

Размер и расположение осей, включая метки, заголовок и поля, указанные как четыре элемента вектор вида `[` ширина нижнего левого угла]. Этот вектор определяет экстенды прямоугольник, который охватывает внешние границы осей. Левый и нижний элементы определить расстояние от нижнего левого угла окна приложения, которое содержит оси в нижний левый угол прямоугольника. Последние два элемента, ширина и высота элементы - это размеры прямоугольника.

5-17

---

## Страница 78

5 [Ссылка на недвижимость](#)

Значения измеряются в единицах, нормализованных к контейнеру. Значение по умолчанию `[0 0 1 1]` включает в себя всю внутреннюю часть контейнера.

На следующем изображении красный прямоугольник охватывает область, определенную `OuterPosition` свойство. Зеленый прямоугольник охватывает область, определенную свойством `Position`.

Для получения дополнительной информации см. Раздел «Изменение размеров осей для размещения названий и меток» в документация MATLAB.

**Родитель** - Родитель осей  
 объект окна приложения

5-18

---

## Страница 79

Свойства осей

Родитель осей, указанный объект окна приложения.

**PlotBoxAspectRatio** - Относительная длина каждой оси  
[1 1 1] | [px py n]

Относительная длина каждой оси, указанная в виде трехэлементного вектора вида [px py n] определяет относительных масштабных коэффициентов оси X и оси Y. Третий векторный элемент игнорируется, но должен присутствовать. График представляет собой блок, охватывающий область данных осей, как определено пределы оси.

Свойство PlotBoxAspectRatio взаимодействует с DataAspectRatio, XLimMode и Свойства YLimMode. Установка PlotBoxAspectRatio отключает «растягивать, чтобы заполнить поведение формы окна приложения, если DataAspectRatioMode и PlotBoxAspectRatioMode оба значения свойства 'auto'.

Если для соответствующего свойства mode установлено значение «auto», то MATLAB выбирает значения отношения. Если вы установите это свойство, то MATLAB установит режим на «ручной».

Пример: [1,0.75,0.75]

Типы данных: один | двойной | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64

**PlotBoxAspectRatioMode** - режим выбора для **PlotBoxAspectRatio**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим выбора для свойства PlotBoxAspectRatio, указанного в качестве одного из следующих значений:

- «Авто» - используйте значения, которые наилучшим образом используют область, предоставленную окном приложения.
- «ручной» - используйте значения, указанные вручную. Чтобы указать значения, установите свойство PlotBoxAspectRatio. Изменение PlotBoxAspectRatioMode на «manual» отключает поведение «растянуть, чтобы заполнить форму окна приложения», если значения свойств PlotBoxAspectRatioMode и DataAspectRatioMode являются 'авто'.

В этой таблице описывается поведение различных комбинаций свойств при отключении поведение при растяжении до заполнения.

XLimitMode и YLimitMode	DataAspectRatioMode	PlotBoxAspectRatioMode Behavior	
'авто'	'авто'	'авто'	Пределы, выбранные для охвата диапазон данных во всех размеры.

XLimitMode и YLimitMode	DataAspectRatioMode	PlotBoxAspectRatioMode Behavior	
'авто'	'авто'	'руководство по эксплуатации'	Выбранные ограничения диапазон данных диапазона во всех измерениях. MATLAB модифицирует DataAspectRatio к достичь запрошенного PlotBoxAspectRatio в пределах программное обеспечение выбрано.
'авто'	'руководство по эксплуатации'	'авто'	Выбранные ограничения диапазон данных диапазона во всех измерениях. MATLAB модифицирует PlotBoxAspectRatio к достичь запрошенного DataAspectRatio в пределах программное обеспечение выбрано.
'авто'	'руководство по эксплуатации'	'руководство по эксплуатации'	Выбранные ограничения полностью подходит и центрировать сюжет

'руководство по эксплуатации' 'авто'	'авто'
--------------------------------------	--------

Страница 81

XLimitMode и YLimitMode	DataAspectRatioMode	PlotBoxAspectRatioMode Behavior
'руководство по эксплуатации' 'авто'		'руководство по эксплуатации' MATLAB отличает пределы и <code>PlotBoxAspectRatio</code> и модифицирует <code>DataAspectRatio</code> as необходимо.
'руководство по эксплуатации' 'руководство по эксплуатации' 'авто'		MATLAB соблюдает ограничения и <code>DataAspectRatio</code> и модифицирует <code>PlotBoxAspectRatio</code> as необходимо.

5-21



Режим выбора для свойства TickDir, указанного в качестве одного из следующих значений:

- 'auto' - использовать направление по умолчанию
- «Вручную» - использовать ручное указание направления галочки. Чтобы указать галочку направление, установите свойство TickDir.

**TickLength** - длина отметки  
[0.01 0.025] (по умолчанию) | [Длина, п]

Длина галочки, указанная в виде двухэлементного вектора вида [длина, п]. Первый Элемент - длина отметки. Второй элемент игнорируется (но должен быть указан). Укажите значения в единицах, нормализованных относительно самой длинной видимой оси X или оси Y линий.

Пример: [0,02 0,035]

**Заголовок** - текстовый объект, используемый для заголовка осей  
текстовый объект (по умолчанию)

Текстовый объект, используемый для заголовка осей. Используйте этот текстовый объект для изменения свойств текста заголовка.

```
window = appwindow;  
ax = axes ( 'Parent' , окно);  
ax.Title.String = 'Заголовок моего графика' ;  
ax.Title.FontWeight = 'normal' ;
```

Примечание: чтобы получить доступ к текстовому объекту заголовка осей, используйте свойство Title или функцию title. Этот текстовый объект не содержится в свойстве Axes Children, его нельзя вернуть

findobj, и он не использует значения по умолчанию, определенные для текстовых объектов.

**Тип** - Тип графического объекта  
'оси' (по умолчанию)

Тип графического объекта, возвращаемого в виде строки «оси».

Типы данных: char

**Единицы** - Позиционные единицы  
«нормализованный» (по умолчанию) | 'дюймов' | сантиметры | «пикселей» | «очки» | 'персонажи'

Позиционные единицы, указанные как одно из значений в следующей таблице.

Единицы	Описание
«нормализовано» (по умолчанию)	Нормализовано относительно контейнера, которая, как правило, фигура или uipanel. Нижний левый угол карты конатинера до (0,0) и в верхнем правом углу карты к (1,1).
«» дюймов	Дюймы.
«» сантиметров	Сантиметры.
'персонажи'	На основе размера символов в системный шрифт по умолчанию. Ширина одного символьная единица - ширина буквы
	Икс. Высота одной символьной единицы расстояние между базовыми линиями двух линий текста.
'точки'	Точки. Одна точка равна 1/72 дюйма.

«пикселей»

Пиксели. Размер пикселя зависит от экрана разрешающая способность.

При указании единиц измерения в качестве имени, пары значений при создании объекта необходимо установить Свойство Units перед указанием свойств, которые вы хотите использовать, например: Позиция.

**Видимый** - Видимость осей  
'оп' (по умолчанию) | «Выключено»

Видимость осей, указанная в качестве одного из следующих значений:

- «вкл.» - отображение осей.
- «выкл» - скрыть оси, не удаляя их. Вы все еще можете получить доступ к свойствам объект невидимые оси.

**XAxisLocation** - Расположение оси X  
«нижний» (по умолчанию) | 'Вверх'

Расположение оси X , указанное в качестве одного из следующих значений:

- «низ» - отображение оси x в нижней части осей

- или «верх» - отображение оси x в верхней части осей

**XColor , YColor** - Цвет контуров осей и отметок  
[0,15 0,15 0,15] (по умолчанию) | RGB триплет | цветная строка | 'никто'

Цвет контуров осей в направлении x или y и соответствующих отметок, как триплет RGB, цветная строка или «нет». Если вы установите цвет «нет», то оси контур невидим.

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне [0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«Y»	[1 1 0]
«Малиновый»	«M»	[1 0 1]
«Голубой»	«C»	[0 1 1]
«Красный»	'r'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«Б»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный	«К»	[0 0 0]

Пример: [1 1 0]

Пример: 'y'

Пример: «желтый»

**XColorMode , YColorMode** - режим выбора цвета контуров осей  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим выбора цвета контура оси, заданный одним из следующих значений:

- «Авто» - использовать цвет по умолчанию.
- «ручной» - используйте цвет, указанный вручную. Чтобы указать цвет, установите XColor или YColor собственность.

См. GridColorMode для связанной информации о выборе цвета линий сетки.

**XDir , YDir** - Направление возрастания значений вдоль оси  
'нормальный' (по умолчанию) | 'задний ход'

Направление увеличения значений вдоль оси, указанной в качестве одного из следующих значений:

- «нормальный» - нормальное направление увеличения значений:
  - Значения оси X увеличиваются слева направо.
  - значения оси Y увеличиваются снизу вверх.
- 'reverse' - обратное направление увеличения значений:
  - Значения оси X уменьшаются слева направо.
  - значения оси Y уменьшаются снизу вверх.

**XGrid , YGrid** - Отображение линий сетки  
'выкл' (по умолчанию) | 'на'

Отображение линий сетки, указанных как одно из следующих значений:

- «Выкл.» - не отображать линии сетки.
- «вкл» - отображает линии сетки перпендикулярно оси, например, вдоль линий постоянные значения x или y. Используйте команду grid, чтобы установить оба свойства «on» или «off».

**XLabel , YLabel** - Текстовый объект  
текстовый объект (по умолчанию)

Текстовый объект для оси X или Y, метка. Используйте этот текстовый объект, чтобы изменить свойства текст метки оси.

```
window = appwindow;
ax = axes ( 'Parent' , окно);
ax.YLabel.String = 'Ось Y' ;
ax.YLabel.FontSize = 12;
```

Примечание. Для доступа к текстовым объектам меток оси используйте XLabel и YLabel, свойства или функции xlabel и ylabel. Эти текстовые объекты не содержатся в осях

Дочерние свойства, они не могут быть возвращены findobj, и они не используют по умолчанию значения, определенные для текстовых объектов.

**XLim , YLim** - минимальные и максимальные пределы оси  
[0 1] (по умолчанию) | [мин Макс]

Минимальные и максимальные пределы по оси X или Y, заданные как двухэлементный вектор форма [мин макс].

Если для соответствующего свойства mode установлено значение 'auto', то MATLAB выбирает пределы оси. Если вы присваиваете значение этому свойству, то MATLAB устанавливает режим на «ручной» и не вычисляет автоматически пределы

**XLimMode , YLimMode** - режим выбора пределов оси  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим выбора пределов оси, указанный в качестве одного из следующих значений:

- «Авто» - выберите пределы оси на основе нанесенных данных. То есть общий промежуток XData или YData всех объектов, отображаемых на осях.
- «Вручную» - использовать указанные вручную пределы оси. Чтобы указать пределы оси, установите Свойство XLim или YLim.

**XMinorGrid , YMinorGrid** - отображение второстепенных линий сетки

'выкл' (по умолчанию) | 'на'

Отображение второстепенных линий сетки, указанных в качестве одного из следующих значений:

- «Выкл.» - не отображать линии сетки.
- «вкл.» - отображение линий сетки, выровненных по второстепенным отметкам оси. Ты не делай необходимо включить второстепенные тики для отображения второстепенных линий сетки.

**XMinorTick , YMinorTick** - отображение незначительных отметок

'выкл' (по умолчанию) | 'на'

Отображение незначительных отметок, указанных как одно из следующих значений:

- «Выкл.» - не отображать незначительные отметки.
- «вкл.» - отображать незначительные отметки между основными отметками на оси. расстояние между основными отметками определяет количество второстепенных отметок.

**XScale , YScale** - шкала значений вдоль оси

«линейный» (по умолчанию) | 'журнал'

Шкала значений вдоль оси, указанная как «линейная» или «лог».

5-26

**XTick , YTick** - Места отметок

[] (по умолчанию) | вектор возрастающих значений

Места отметки отмечены как вектор возрастающих значений.

Если для соответствующего свойства mode установлено значение 'auto', то MATLAB выбирает значения тиков. Если вы присваиваете значение этому свойству, то MATLAB устанавливает режим на «ручной» и автоматически не пересчитывает значения тиков.

Пример: [2 4 6 8 10]

Пример: 0: 10: 100

Типы данных: один | двойной

**XTickLabel , YTickLabel** - метки для отметок

" (по умолчанию) | массив ячеек строк

Метки меток, заданные в виде массива ячеек строк. Если вы не указали достаточно строк для всех отметок тиков, то оси циклически проходят указанные строки.

Если для соответствующего свойства mode установлено значение «auto», то MATLAB выбирает метки. Если Вы присваиваете значение этому свойству, затем MATLAB устанавливает режим на «ручной» и делает не автоматически выбирать метки.

**XTickLabelMode , YTickLabelMode** - режим выбора меток для отметок

'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим выбора меток для отметок, указанный как одно из следующих значений:

- 'auto' - помечать метки галочками с числовыми значениями, которые охватывают диапазон нанесенного графика данные.
- «ручной» - использовать метки, помеченные вручную. Чтобы указать метки, установите Свойство XTickLabel или YTickLabel.

**XTickMode , YTickMode** - режим выбора местоположения отметок

'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим выбора местоположения отметок, заданный одним из следующих значений:

- «Авто» - выберите расположение меток на основе диапазона данных для оси.

- «Вручную» - использовать вручную отмеченные места для отметок. Чтобы указать значения, установите свойство XTick или YTick.

5 Ссылка на недвижимость

**YAxisLocation** - Расположение оси Y  
'left' (по умолчанию) | 'правильно'

Расположение оси Y , указанное в качестве одного из следующих значений:

- 'left' - отображение оси Y на правой стороне осей.

\

'right' - отображение оси Y на левой стороне осей.

Смотрите также  
оси | ось | коробка | каксис | кла | GCA | сетка

Больше о  
• «Доступ к стоимости недвижимости»

## Свойства кнопки

Контроль внешнего вида и поведения кнопок

Кнопки - это компоненты приложения, которые обычно генерируют действие при нажатии пользователем приложения. и выпускает их. Свойства кнопки управляют внешним видом и поведением конкретного экземпляра кнопки. Чтобы изменить аспекты кнопки, измените значения свойств. Используйте точку запись для обозначения конкретного объекта и свойства:

```
кнопка = uibutton;
txt = button.Text;
button.Text = 'Plot';
```

**ButtonPressedFcn** - код для выполнения после того, как пользователь приложения нажмет и **отпустит** кнопку  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Код для выполнения, когда пользователь приложения нажимает и отпускает кнопку, указанную в качестве функции handle, массив ячеек, содержащий функцию handle и дополнительные аргументы, или строку.

MATLAB не выполняет функцию обратного вызова, если пользователь приложения нажимает кнопку, но затем перемещает мышь и отпускает кнопку мыши, когда курсор находится за пределами границы кнопки.

Пример: @function

Пример: {@function, x}

**Включено** - рабочее состояние кнопки

true (по умолчанию) | ложный

Рабочее состояние кнопки, заданное как true или false.

Если для этого свойства установлено значение true, пользователь приложения может нажать кнопку.

Если для этого свойства установлено значение false, кнопка будет недоступна, что означает, что приложение пользователь не может нажать ее, и кнопка не будет вызывать обратный вызов.

**FontAngle** - символьный наклон текста кнопки

'нормальный' (по умолчанию) | «Наклонным»

5-29

## Страница 90

5 Ссылка на недвижимость

Символьный уклон текста кнопки, указанный как «обычный» или «курсив». Установка этого свойства курсивом выбирает наклонную версию шрифта, если она доступна пользователю приложения система.

**FontName** - шрифт для отображения текста кнопки

'Helvetica' (по умолчанию) | строка

Шрифт, в котором отображается текст кнопки, указанный в виде строки. Если указанный шрифт недоступно пользователю приложения, MATLAB использует Helvetica. Если Helvetica не доступно, MATLAB использует доступный шрифт без засечек.

Пример: «Ариал»

**FontSize** - размер шрифта текста кнопки

12 (по умолчанию) | положительное число

Размер шрифта текста кнопки, указанный как положительное число. Единицы являются точками.

Пример: 14

Типы данных: двойной

**FontWeight** - толщина текстовых символов

'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**HorizontalAlignment** - Горизонтальное выравнивание значка и текста внутри кнопки

'центр' (по умолчанию) | «левый» | 'правильно'  
Горизонтальное выравнивание текста и значка внутри кнопки, обозначенных как «центр», «слева», или вправо'.  
  
На следующем рисунке показаны три варианта горизонтального выравнивания: по центру, слева и справа, когда для свойств VerticalAlignment и IconAlignment установлены значения по умолчанию ('center' и 'left' соответственно).

5-30

**Icon** - имя файла иконки для отображения на кнопке  
" (по умолчанию) | строка  
  
Имя файла иконки для отображения на кнопке, указанной в виде строки.  
  
Строка должна быть именем и расширением файла изображения в пути MATLAB, или полный путь к файлу изображения.  
  
Тип файла изображения должен быть JPEG, GIF или PNG.  
  
MATLAB обрезает изображение, если не уместается в пределах, указанных кнопкой Размер  
стоимость имущества.  
  
Пример: 'icon.png'  
  
Пример: 'C: \ Documents \ icon.png'

**IconAlignment** - расположение значка кнопки относительно текста  
'left' (по умолчанию) | «центр» | «верх» | «дно» | 'правильно'  
  
Расположение значка кнопки относительно кнопки «Текст» (если есть), указанного как одно из следующих  
ценности:

Стоимость	Результат
'оставил'	
'центр'	
'Вверх'	
'низ'	
'правильно'	

5-31

5 Ссылка на недвижимость

Примечание. Предыдущие изображения отражают расположение значков при горизонтальном выравнивании и  
Каждому VerticalAlignment присваивается значение по умолчанию, «center».

Если вы укажете пустую строку (") для свойства Text, то значок будет выровнен, как если бы это текст, в соответствии со значениями, установленными для HorizontalAlignment и VerticalAlignment. Значение свойства IconAlignment в этом случае не влияет.

**Расположение** - расположение кнопки относительно родителя  
[100,100] (по умолчанию) | [X, y]

Расположение кнопки относительно родителя, заданное как [x, y], где x и y определяют координаты от нижнего левого угла родительского контейнера до нижнего левого угла кнопка. Единицами измерения являются пиксели.

**OuterLocation** - расположение кнопки относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для кнопок.

**OuterSize** - размер кнопки, включая значение текста  
[100,20] (по умолчанию) | [ширина высота]

Размер кнопки, возвращается как [ширина, высота]. Единицами измерения являются пиксели.

**Родитель** - Родительский контейнер кнопки  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер кнопки, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - размер кнопки  
[100,20] (по умолчанию) | [ширина высота].

Размер кнопки, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Текст** - символы, отображаемые на кнопке  
«Кнопка» (по умолчанию) | строка | массив ячеек строк

Символы, отображаемые на кнопке, указаны как одно из следующих:

- строка - отображается в виде одной строки. Например:

```
uibutton ( «Текст» , «Пресса» );
```

- массив строк - отображается как многострочный текст. Каждая строка массива ячеек строка текста. Отрегулируйте значение свойства Размер кнопки соответственно.

```
uibutton ( 'Text' , { 'Press' , 'Here' } , 'Size' , [40,40]);
```

Пример: «Перезапустить симуляцию»

Пример: {'Restart', 'Simulation'}

**VerticalAlignment** - вертикальное выравнивание текста внутри кнопки  
'центр' (по умолчанию) | «верх» | 'низ'

Вертикальное выравнивание текста внутри кнопки, указанной как «центр», «верх» или 'низ'.

На следующем рисунке показаны три варианта вертикального выравнивания, когда Горизонтальное выравнивание является «центром».



Примечание. Значения вертикального выравнивания не действуют, если кнопка используется по умолчанию высота.

**Видимый** - Видимость кнопки  
true (по умолчанию) | ложный

5-33

Страница 94

5 Ссылка на недвижимость

Видимость кнопки, указанная как true или false. Свойство Visible определяет, является ли кнопка отображается на экране. Если для свойства Visible кнопки установлено значение false, вся кнопка скрыта, но вы все равно можете указать и получить доступ к ее свойствам.

5-34

Страница 95

Свойства группы кнопок

## Свойства группы кнопок

Группа кнопок управления внешним видом и поведением

Группы кнопок - это компоненты приложения, которые управляют взаимной исключительностью радио кнопки или кнопки переключения, которые он содержит. Группа кнопок может содержать переключатели или кнопки переключения, но не оба. Как правило, группа кнопок генерирует действие, когда выбранная кнопка изменится. Свойства контролируют внешний вид и поведение конкретного экземпляра группы кнопок. Чтобы изменить аспекты группы кнопок, измените значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
window = appwindow;
bg = matlab.ui.control.ButtonGroup («Родитель», окно);
title = bg.Title;
bg.Title = 'Опции';
```

**BorderVisibility** - **отображается** ли граница группы кнопок  
true (по умолчанию) | ложный

Отображается ли граница группы кнопок, указанная как true или false.

**Включено** - рабочее состояние группы кнопок  
true (по умолчанию) | ложный

Рабочий статус группы кнопок, указанный как true или false.

Если для этого свойства установлено значение true, внешний вид группы кнопок означает, что пользователь приложения может взаимодействовать с дочерними компонентами (если вы не отключите отдельные компоненты внутри группа кнопок).

Если для этого свойства установлено значение false, группа кнопок будет затемнена, что означает, что Пользователь приложения не может взаимодействовать с дочерними компонентами, и группа кнопок не будет вызывать Перезвони.

5-35

### Страница 96

5 Ссылка на недвижимость

Примечание Изменения Enabled свойства кнопки группы никак *не* меняют значения свойства Enabled его дочерних компонентов, даже при отключении группы кнопок заставляет детей быть инвалидами.

**FontAngle** - символьный наклон заголовка группы кнопок  
'нормальный' (по умолчанию) | «Наклонным»

Символьный уклон заголовка группы кнопок, значение указано как «обычный» или «курсив».

При установке этого свойства курсивом выбирается наклонная версия шрифта, если она доступна на система пользователя приложения.

**FontName** - шрифт, в котором отображается название группы кнопок  
'Helvetica' (по умолчанию) | строка

Шрифт для отображения группы кнопок Значение свойства заголовка, заданное в виде строки.

Если указанный шрифт недоступен для пользователя приложения, MATLAB использует Helvetica. Если

Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

Пример: «Ариал»

**FontSize** - размер шрифта заголовка группы кнопок  
12 (по умолчанию) | положительное число

Размер шрифта заголовка группы кнопок, указанный как положительное число. Единицы измерения являются точками.

Пример: 14

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальному весу шрифта.

**Расположение** - расположение группы кнопок относительно родителя  
[100,100] (по умолчанию) | [X, y]

5-36

Расположение группы кнопок относительно родителя, заданное как [x, y], где x и y определяют координаты от нижнего левого угла родительского контейнера до нижнего левого угла группа кнопок. Единицами измерения являются пиксели.

**OuterSize** - размер группы кнопок, включая строку заголовка  
[123,106] (по умолчанию) | [ширина высота]

Размер группы кнопок, включая строку заголовка, возвращается как [ширина, высота]. Единицы измерения являются пикселями.

**Parent** - Родительский контейнер группы кнопок  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер группы кнопок, указанный как окно приложения, панель или группа кнопок объект.

**SelectedObject** - выбранное в настоящий момент радио или кнопка переключения  
первое радио или переключатель в группе кнопок (по умолчанию)

Текущий выбранный переключатель или переключатель, указанный в качестве переключателя или переключателя объект кнопки.

Получите значение этого свойства, чтобы определить, какая радио или кнопка переключения в настоящее время выбран в группе кнопок.

Задайте значение этого свойства, чтобы изменить текущее выбранное радио или кнопку переключения. Когда вы изменяете выделение с помощью этого свойства, MATLAB настраивает свойство Value для другого радио или переключателя кнопок в группе кнопок соответственно.

Например, предположим, что ваша группа кнопок содержит три переключателя, и вы установили Свойство SelectedObject для radiobutton3. MATLAB устанавливает свойство Value для каждого Дочерняя кнопка радио следующим образом:

- radiobutton1.Value = false;
- radiobutton2.Value = false;
- radiobutton3.Value = true;

Другими словами, установка свойства SelectedObject для группы кнопок имеет тот же эффект как установка свойства Value для дочерней кнопки.

**SelectionChangeFcn** - функция, которая выполняется, когда пользователь приложения меняет кнопку выбор

[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция, которая выполняется, когда пользователь приложения меняет выбор кнопки, указанный как один из эти значения:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области

Эта функция обратного вызова не выполняется, если изменяется значение переключателя или переключателя. программно.

Пример: @function

Пример: {@function, x}

**Размер** - размер группы кнопок, исключая строку заголовка [123,85] (по умолчанию) | [ширина высота]

Размер группы кнопок, исключая строку заголовка, указывается как [ширина, высота]. Единицы измерения являются пикселями.

**Заголовок** - название группы кнопок «Группа кнопок» (по умолчанию) | строка | массив ячеек строк

Название группы кнопок, указанное в виде строки.

**Видимый** - Видимость группы кнопок true (по умолчанию) | ложный

Видимость группы кнопок, указанная как true или false. Видимое свойство определяет отображается ли группа кнопок на экране. Если свойство Visible кнопки Для группы установлено значение false, вся группа кнопок скрыта. Тем не менее, вы все равно можете указать и получить доступ к его свойствам.

Примечание Изменения свойства Видимых кнопок группы никак *не* изменят значения Видимые свойства его дочерних компонентов, даже если скрытие группы кнопок вызывает дети должны быть спрятаны.

## Флажок Свойства

Контроль флажка внешнего вида и поведения

Флажки - это компоненты приложения, которые указывают одно из двух состояний. Как правило, флажки генерировать действие при проверке или очистке. Свойства контролируют внешний вид и поведение конкретного экземпляра флажка. Чтобы изменить аспекты флажка, измените значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
checkbox = uicontrol;  
txt = checkbox.Text;  
checkbox.Text = 'Показать значение';
```

**Включено** - Флажок рабочего состояния true (по умолчанию) | ложный

Рабочее состояние флажка, указанного как true или false.

Если для этого свойства установлено значение true, пользователь приложения может выбрать или очистить его.

Если для этого свойства установлено значение false, флажок будет недоступен, что означает, что приложение пользователь не может выбрать или очистить его, и он не будет вызывать обратный вызов.

**FontAngle** - символьный уклон текста флажка

'нормальный' (по умолчанию) | «Наклонным»

Наклон символа текста флажка, указанного как «обычный» или «курсив». Установка этого свойства курсивом выбирает наклонную версию шрифта, если она доступна пользователю приложения система.

**FontName** - шрифт, в котором отображается текст флажка

'Helvetica' (по умолчанию) | строка

Шрифт для отображения флажка Значение свойства текста, заданное в виде строки. Если указанный шрифт недоступен пользователю приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

**FontSize** - размер шрифта текста флажка

12 (по умолчанию) | положительное число

5-39

## Страница 100

5 Ссылка на недвижимость

Размер шрифта текста флажка, указанного как положительное число. Единицами измерения являются точки.

Пример: 14

**FontWeight** - толщина текстовых символов

'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**Расположение** - расположение флажка относительно родительского контейнера

[100,100] (по умолчанию) | [X, y]

Расположение флажка относительно родительского контейнера, указанного как [x, y], где x и y укажите координаты в левом нижнем углу родительского контейнера (панели или приложения) окно) в нижнем левом углу флажка. Единицами измерения являются пиксели.

Пример: [100,150]

**OuterLocation** - Расположение флажка относительно родительского контейнера.

[100,100] (по умолчанию) | [X, y]

Идентично местоположению для флажков.

**OuterSize** - Размер флажка, включая текстовое значение

[84,15] (по умолчанию) | [ширина высота]

Идентичен по размеру для флажков.

**Родитель** - Родительский контейнер флажка

объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер флажка, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - размер флажка, включая значение текста

[84,15] (по умолчанию) | [ширина высота]

## Страница 101

## Флажок Свойства

Размер флажка, включая значение «Текст», задается как [ширина, высота]. Единицы измерения являются пикселями.

**Текст** - флажок

«Флажок» (по умолчанию) | строка | массив ячеек строк

Флажок флажка, указанный как один из следующих:

- строка - отображается в виде одной строки. Например:

```
uicontrol ( 'Текст' , «Фильм» );
```

- массив строк - отображается как многострочный текст. Каждая строка массива ячеек строка текста. (Вы должны увеличить размер от значения по умолчанию, чтобы вместить дополнительная строка текста.)

```
uicontrol ( 'Text' , { 'Notch' , 'Filter' } , 'Size' , [84,30]);
```

Если вы установите свойство Text для массива ячеек строк, вы должны изменить флажок Размер значение свойства для размещения дополнительных строк текста.

**Значение** - состояние флажка

false (по умолчанию) | правда

Состояние флажка указано как истинное или ложное. Когда для свойства Value установлено значение true, флажок установлен; если установлено значение false, флажок не установлен.

**ValueChangedFcn** - функция, которая выполняется, когда пользователь приложения устанавливает или снимает флажок

[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция, которая выполняется, когда пользователь приложения устанавливает или снимает флажок, указанный как один из эти значения:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Эта функция обратного вызова не выполняется, если флажок Значение изменяется программно.

Пример: @function

Пример: {@function, x}

**Видимый** - Флажок видимости

true (по умолчанию) | ложный

## Страница 102

5 Ссылка на недвижимость

Флажок видимости, указанный как true или false. Видимое свойство определяет отображается ли флажок на экране. Если видимое свойство флажка установлен в false, весь флажок скрыт, но вы все равно можете указать и получить доступ к его свойства.

5-42

---

Страница 103

Свойства дискретной ручки

## Свойства дискретной ручки

Управление внешним видом и поведением дискретной ручки

Дискретные ручки - это компоненты приложения, из которых пользователь может выбрать один вариант из набора.

Как правило, дискретные ручки генерируют действие, когда пользователь меняет выбор ручки.

Свойства управляют внешним видом и поведением конкретного экземпляра

регулятор. Чтобы изменить аспекты дискретной ручки, измените значения свойств. Используйте точечную запись для ссылаются на конкретный объект и свойство:

```
knob = uiknob («дискретный»);  
val = knob.Value;  
knob.Value = 'Low';
```

**Включено** - рабочее состояние дискретной ручки

true (по умолчанию) | ложный

Рабочее состояние дискретной ручки, заданное как true или false.

Если для этого свойства установлено значение true, пользователь приложения может поворачивать дискретную ручку.

Если вы установите для этого свойства значение false, внешний вид дискретной ручки будет затемнен, указывает на то, что пользователь приложения не может повернуть дискретную ручку, и он не будет вызывать Перезвони.

**Location** - расположение кнопки, исключая государственные знаки и метки относительно родительского

[126,111] (по умолчанию) | [X, y]

Расположение дискретной ручки, исключая государственные метки и метки, относительно родительского, указано как [x, y], где x и y определяют координаты из левого нижнего угла родителя

Контейнер в нижнем левом углу дискретной ручки. Единицами измерения являются

пиксели.

**OuterLocation** - расположение ручки, включая метки состояния и метки относительно родительского [101,110] (по умолчанию) | [X, Y]

5-43

5 Ссылка на недвижимость

Расположение ручки, включая отметки, относительно родителя, указано как [x, y]. Единицы измерения являются пикселями.

**OuterSize** - размер ручки, включая государственные метки и метки [127,86] (по умолчанию) | [высота ширина]

Размер ручки, включая метки состояния и метки, возвращается как [высота, ширина]. Единицы измерения являются пикселями.

**Parent** - родительский контейнер с дискретной ручкой  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер с дискретной ручкой, указанный как окно приложения, панель или группа кнопок объект.

**Размер** - размер кнопки, без учета государственных марок и этикеток [60,60] (по умолчанию) | [ширина высота]

Размер ручки, без учета государственных знаков и меток, определяется как [ширина, высота]. Единицами измерения являются пиксели.

**Текст** - набор меток дискретных ручек  
{'Выкл.', 'Низкий', 'Средний', 'Высокий'} (по умолчанию) | Массив строк размером 1 на n  
Набор меток дискретных ручек, заданный в виде массива строк размером 1 на n. Массив ячеек должен содержать как минимум два элемента. Дискретная ручка отображает столько вариантов, сколько есть являются элементами в значении свойства Text. MATLAB помечает метки ручек, используя значения массива ячеек, примененные по часовой стрелке к ручке.

Пример: {'Off', 'Slow', 'Fast'}

Пример: {'1', '2', '3', '4'}

**TextData** - данные, связанные с каждым элементом свойства **Text** {1,2,3,4} (по умолчанию) | Ячейка целых чисел размером 1 на n

Данные, связанные с каждым элементом свойства Text, заданные как ячейка 1 на n массив целых чисел, где n больше 0. Например, если вы установите свойство Text в описания температуры («кипение», «тепло», «заморозание»), вы можете установить Свойство TextData для соответствующих значений температуры ({212, 90, 32} - в градусах в градусах Фаренгейта).

Если вы измените количество элементов в значении свойства Text, тогда TextData свойство обновляется соответствующим образом, *если* вы ранее явно не установили TextData. если ты

5-44

Свойства дискретной ручки

явно установить свойство TextData, тогда оно не обновляется автоматически, когда количество элементов в свойстве Text изменяется.

Если значение свойства TextData содержит больше элементов, чем значение свойства Text, затем MATLAB игнорирует дополнительные элементы значения свойства TextData.

Если значение свойства TextData содержит меньше элементов, чем значение свойства Text,



и пользователь приложения выбирает параметр, который не имеет соответствующего элемента свойства TextData, затем MATLAB устанавливает свойство ValueData в пустой массив ячеек ({} ) и MATLAB отображает предупреждение в командной строке.

**Значение** - значение, на которое указывает ручка элемент массива свойств текста

Значение, на которое указывает дискретная ручка, указанное в качестве элемента свойства Text массив. По умолчанию это первый элемент массива свойств Text.

Если массив свойств Text пуст, MATLAB устанавливает для свойства Value пустое значение типа double. массив [].

Если массив свойств текста изменяется программно, а текущим значением является элемент массива свойств текста, то значение остается неизменным.

Если массив свойств текста изменяется программно и значение параметра больше не элемент массива свойств текста, MATLAB устанавливает значение первого элемента Text property array.

Если свойство ValueData изменяется программно, MATLAB обновляет значение свойство элемента массива свойства Text, соответствующее новому выделению.

Если свойство Value изменяется программно на элемент в ValueData, который появляется более одного раза в ValueData, затем элемент, выбранный в приложении, соответствует первый матч.

**ValueChangedFcn** - обратный **вызов** для выполнения, когда пользователь приложения меняет расположение элемента управления ручки [] (по умолчанию) | ручка функции | массив ячеек | строка

Обратный вызов для выполнения, когда пользователь приложения изменяет настройку управления дискретной ручкой, указано как одно из следующих:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы

5-45

## Страница 106

5 Ссылка на недвижимость

- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пример: @function

Пример: {@function, x}

**ValueData** - данные, связанные с выбранным значением элемент TextData

Данные, связанные с выбранным значением, указываются в качестве элемента TextData стоимость имущества. MATLAB использует isequal для определения значения свойства ValueData соответствует элементу значения свойства TextData. Если TextData пусто, MATLAB устанавливает свойство Value для пустого двойного массива [].

Если вы программно установите значение свойства ValueData, то свойство Value настройки и выбор в приложении меняются соответственно.

Если вы программно установите ValueData для элемента в TextData, который появляется больше чем один раз в TextData, то элемент, выбранный в приложении, соответствует первому совпадению.

Если вы программно измените TextData, текущий выбор в приложении не менять. Однако ValueData обновляется до новых данных выбранной опции.

Если параметр свойства Value изменяется, значение свойства ValueData обновляется до данные, связанные с новым значением.

**Видимый** - видимость дискретной ручки true (по умолчанию) | ложный

Дискретная ручка видимости, указанная как true или false. Видимое свойство определяет отображается ли на экране дискретная ручка. Если видимое свойство дискретного регулятор имеет значение false, вся дискретная ручка скрыта, но вы все равно можете указать и получить доступ к его свойствам.

## Drop Down Properties

Контроль выпадающего внешнего вида и поведения

Раскрывающиеся элементы - это компоненты приложения, из которых пользователь может выбрать один вариант из набора. До пользователь щелкает компонент, отображается только выбранная в данный момент опция. Как правило, выпадающие списки генерируют действие, когда пользователь меняет выбор. Свойства контролируют Внешний вид и поведение конкретного экземпляра раскрывающегося списка. Чтобы изменить аспекты раскрыть, изменить значения свойств. Используйте точечную запись для обозначения конкретного объекта и свойство:

```
раскрывающийся список = uidropdown;  
txt = dropdown.Text;  
dropdown.Text = {'Coccus', 'Bacillus', ...  
                «Риккетсия», «Спириллиум»};
```

**Включено** - рабочее состояние выпадающего  
true (по умолчанию) | ложный

Рабочее состояние раскрывающегося списка, заданное как true или false.

Если для этого свойства установлено значение true, появление раскрывающегося списка означает, что приложение Пользователь может открыть и сделать выбор из выпадающего списка.

Если для этого свойства установлено значение false, раскрывающийся список будет затемнен, указывает, что пользователь приложения не может открыть или сделать выбор из выпадающего списка, и это не будет вызывать никаких обратных вызовов.

**FontAngle** - наклон символов выпадающего текста  
'нормальный' (по умолчанию) | «Наклонным»

Символьный наклон выпадающего текста, указанного как «обычный» или «курсив». Установка этого свойство курсивом выбирает наклонную версию шрифта, если она доступна пользователю приложения система.

'Helvetica' (по умолчанию) | строка

Шрифт для отображения раскрывающегося значения свойства Text, указанного в виде строки. Если указанный шрифт недоступен пользователю приложения, MATLAB использует Helvetica. Если

Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

**FontSize** - размер шрифта выпадающего текста  
12 (по умолчанию) | положительное число

Размер шрифта выпадающего текста, указанный как положительное число. Единицы измерения являются точками.

Пример: 14

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**Расположение** - расположение раскрывающегося списка относительно родителя  
[100,100] (по умолчанию) | [X, Y]

Расположение раскрывающегося списка относительно родителя, указанного как [x, y], где x и y указывают координаты от нижнего левого угла родительского контейнера до нижнего левого угла закрытый выпадающий. Единицами измерения являются пиксели.

**OuterLocation** - расположение раскрывающегося **списка** относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, Y]

Идентично местоположению для выпадающих меню.

**OuterSize** - Размер выпадающего  
[100, 20] (по умолчанию) | [ширина высота]

Идентичный по размеру для выпадающих.

**Родитель** - Родительский контейнер выпадающего  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер раскрывающегося списка, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - Размер выпадающего  
[100,20] (по умолчанию) | [ширина высота].

Размер раскрывающегося списка, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Текст** - набор раскрывающихся опций, представленных пользователю приложения  
{«Вариант 1», «Вариант 2», «Вариант 3», «Вариант 4»} (по умолчанию) | 1-я ячейка массива строки

Набор выпадающих меток, заданный в виде массива строк размером 1 на n. Выпадающий отображает столько параметров, сколько есть элементов в значении свойства Text.

Пример: {'Красный', 'Желтый', 'Синий'}

Пример: {'1', '2', '3'}

**TextData** - набор раскрывающихся значений  
{1, 2, 3, 4} (по умолчанию) | Ячейка размером 1 на n

Данные, связанные с каждым элементом свойства Text, заданные как массив ячеек размером 1 на n целых чисел, где n больше 0. Например, если вы установите свойство Text в имена сотрудников, вы можете установить для свойства TextData соответствующий идентификатор сотрудника

номера.

Если вы измените количество элементов в значении свойства Text, тогда TextData свойство обновляется соответствующим образом, *если* вы ранее явно не установили TextData. если ты явно установить свойство TextData, тогда оно не обновляется автоматически, когда количество элементов в свойстве Text изменяется.

Если значение свойства TextData содержит больше элементов, чем значение свойства Text, затем MATLAB игнорирует дополнительные элементы значения свойства TextData.

Если значение свойства TextData содержит меньше элементов, чем значение свойства Text, и пользователь приложения выбирает параметр, который не имеет соответствующего элемента свойства TextData, затем MATLAB устанавливает свойство ValueData в пустой массив ячеек ({}). MATLAB отображает предупреждение в командной строке.

Пример: {'Красный' 'Зеленый' 'Синий'}

5-49

## Страница 110

5 Ссылка на недвижимость

Пример: {10 20 30 40}

Пример: {true false}

**Значение** - выбранное значение  
элемент массива свойств текста Text | строка

Выбранное значение, указанное в качестве элемента массива свойства Text.

По умолчанию это первый элемент массива свойств Text.

Если массив свойств Text пуст, MATLAB устанавливает для свойства Value пустое значение типа double. массив [].

Если массив свойств текста изменяется программно, а текущим значением является элемент массива свойств текста, то значение остается неизменным.

Если массив свойств текста изменяется программно и значение параметра больше не элемент массива свойств текста, MATLAB устанавливает значение первого элемента Текстовый массив свойств.

Если свойство ValueData изменяется программно, MATLAB обновляет значение свойство элемента массива свойства Text, соответствующее новому выделению.

Если свойство Value изменяется программно на элемент в ValueData, который появляется более одного раза в ValueData, затем элемент, выбранный в приложении, соответствует первый матч.

**ValueChangedFcn** - код, который нужно выполнить, когда выпадающий выбор пользователя изменится.

[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова для выполнения, когда пользователь приложения меняет выпадающий выбор, указывается как дескриптор функции, массив ячеек, содержащий дескриптор функции и дополнительные аргументы или строка.

Этот обратный вызов не выполняется, если пользователь приложения щелкает выбранную в данный момент опцию, или если значение свойства Value изменяется программно.

Пример: @function

Пример: {@function, x}

**ValueData** - данные, связанные с выбранным значением  
элемент TextData

5-50

## Страница 111

Данные, связанные с выбранным значением, указываются в качестве элемента `TextData` стоимости имущества. MATLAB использует `isequal` для определения значения свойства `ValueData` соответствует элементу значения свойства `TextData`. Если `TextData` пусто, MATLAB устанавливает свойство `Value` для пустого двойного массива `[]`.

Если вы программно установите значение свойства `ValueData`, то свойство `Value` настройки и выбор в приложении меняются соответственно.

Если вы программно установите `ValueData` для элемента в `TextData`, который появляется больше чем один раз в `TextData`, то элемент, выбранный в приложении, соответствует первому совпадению.

Если вы программно измените `TextData`, текущий выбор в приложении не менять. Однако `ValueData` обновляется до новых данных выбранной опции.

Если параметр свойства `Value` изменяется, значение свойства `ValueData` обновляется до данные, связанные с новым значением.

**Видимый** - выпадающий видимость  
true (по умолчанию) | ложный

Выпадающая видимость, указанная как true или false. Видимое свойство определяет отображается ли раскрывающийся список на экране. Если свойство `Visible` выпадающего имеет значение false, весь раскрывающийся список скрыт, но вы все равно можете указать и получить доступ к его свойства.

5-51

## Изменить свойства поля

Контроль внешнего вида и поведения поля редактирования

Поля редактирования - это компоненты приложения, в которые пользователи приложения могут вводить текст. Как правило, редактировать поля генерировать действие, когда пользователь добавляет или изменяет текст. Свойства контролируют внешний вид и поведение конкретного экземпляра поля редактирования. Чтобы изменить аспекты поля редактирования, изменить значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
txtfield = uieditfield;
txt = txtfield.Value;
txtfield.Value = 'Penicillin';
```

**Включено** - рабочее состояние поля редактирования  
true (по умолчанию) | ложный

Рабочее состояние поля редактирования, заданное как true или false.

Если для этого свойства установлено значение true, поле редактирования принимает ввод.

Если вы установите для этого свойства значение false, поле редактирования будет затемнено, что означает, что не принимать ввод и поле редактирования не будет вызывать обратный вызов.

Как показано в следующей таблице, значение свойства Editable также влияет на то, поле работает.

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'правда'		Пользователь приложения может обновить поле редактирования. обратный вызов связан с полем редактирования выполняется, когда текст изменения.
'ложный'	'ложный'		Пользователь приложения не может обновить поле редактирования.

5-52

Страница 113

Изменить свойства поля

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'ложный'		Пользователь приложения не может обновить поле редактирования.  Используйте эту комбинацию стоимости имущества когда ты хочешь текст должен быть легко читаемым, хотя это не редактируемые.
'ложный'	'правда'		Пользователь приложения не может обновить поле редактирования.

**Редактируемый** - редактируется ли поле редактирования true (по умолчанию) | ложный

Является ли поле редактирования доступным для редактирования, указано как true или false.

Как показано в следующей таблице, значение свойства Enabled также влияет на поле редактируемое

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'правда'		Пользователь приложения может обновить поле редактирования. обратный вызов связан с полем редактирования выполняется, когда текст изменения.
'ложный'	'ложный'		Пользователь приложения не может обновить поле редактирования.
'правда'	'ложный'		Пользователь приложения не может обновить поле редактирования.  Используйте эту комбинацию стоимости имущества когда ты хочешь текст должен быть легко читаемым, хотя это не редактируемые.

Включено	Редактирование	Визуальный результат	Функциональный результат
'ложный'	'правда'		Пользователь приложения не может обновить поле редактирования.

**FontAngle** - символьный уклон текста поля редактирования  
'нормальный' (по умолчанию) | «Наклонным»

Символьный уклон текста поля редактирования, указанного как «обычный» или «курсив». Установка этого свойство курсивом выбирает наклонную версию шрифта, если она доступна пользователю приложения система.

**FontName** - шрифт для отображения текста поля редактирования  
'Helvetica' (по умолчанию) | строка

Шрифт, в котором отображается поле редактирования Значение свойства текста, заданное в виде строки. Если указанный шрифт недоступен пользователю приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

**FontSize** - размер шрифта текста поля редактирования  
12 (по умолчанию) | положительное число

Размер шрифта текста поля редактирования, указанный как положительное число. Единицами измерения являются точки.

Пример: 14

Типы данных: двойной

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**HorizontalAlignment** - Горизонтальное выравнивание текста в поле редактирования  
'left' (по умолчанию) | «правильно» | 'центр'

5-54

Выравнивание текста внутри поля редактирования, указанного как «влево», «вправо» или «по центру». выравнивание влияет на отображение, так как пользователь приложения редактирует поле редактирования.

**Расположение** - расположение поля редактирования относительно родительского  
[100,100] (по умолчанию) | [X, y]

Расположение поля редактирования относительно родителя, указанное как [x, y], где x и y задают координаты от нижнего левого угла родительского контейнера до нижнего левого угла поле редактирования. Единицами измерения являются пиксели.

**OuterLocation** - расположение поля редактирования относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для полей редактирования.

**OuterSize** - Размер поля редактирования  
[100,20] (по умолчанию) | [ширина высота]

Идентично размеру для полей редактирования.

**Родитель** - Родительский контейнер поля редактирования  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер поля редактирования, указанный в качестве дескриптора, панели или кнопки окна приложения групповой объект.

**Размер** - Размер поля редактирования  
[100,20] (по умолчанию) | [ширина высота].

Размер поля редактирования, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Значение** - текст в текстовом поле

" (по умолчанию) | строка

Текст в поле редактирования, указанный в виде строки. MATLAB отображает строку в виде одной строки.

Если вы хотите разрешить несколько строк текста, вместо этого создайте текстовую область.

Пример: «Е. палочка»

**ValueChangedFcn** - обратный вызов для выполнения после ввода текста или обновления

[] (по умолчанию) | ручка функции | массив ячеек | строка

5-55

## Страница 116

5 Ссылка на недвижимость

Функция обратного вызова для выполнения после ввода или обновления текста, указанная в качестве одного из следующих значений:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Обратный вызов выполняется после того, как пользователь приложения изменит текст в поле редактирования, а затем либо нажимает **Enter** или щелкает за пределами поля редактирования.

Пример: @function

Пример: {@function, x}

**Видимый** - Редактировать видимость поля

true (по умолчанию) | ложный

Редактировать поле видимости, заданное как true или false. Видимое свойство определяет отображается ли поле редактирования на экране. Если свойство Visible поля редактирования имеет значение false, все поле редактирования скрыто, но вы все равно можете указать и получить доступ к его свойства.



## Редактируемые раскрывающиеся свойства

Управление редактируемым выпадающим внешним видом и поведением

Редактируемые раскрывающиеся списки - это компоненты приложения, из которых пользователь может выбрать один вариант из установить или ввести строку. Пока пользователь не нажмет стрелку вниз, только выбранный в настоящее время опция отображается. Как правило, редактируемые раскрывающиеся списки генерируют действие, когда пользователь меняет выбор. Свойства контролируют внешний вид и поведение конкретного экземпляра редактируемого раскрывающегося списка. Чтобы изменить аспекты редактируемого раскрывающегося списка, измените значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
edropdown = uidropdown («редактируемый»);
txt = edropdown.Text;
edropdown.Text = {'Coccus', 'Bacillus', ...
                 «Риккетсия», «Спирилиум»};
```

**Включено** - работает ли редактируемый раскрывающийся список true (по умолчанию) | ложный

Работоспособен ли редактируемый выпадающий список, заданный как true или false.

Если для этого свойства установлено значение true, вид редактируемого раскрывающегося списка будет ярким.

Если для этого свойства установлено значение false, появится вид редактируемого раскрывающегося списка. серый цвет.

Как показано в следующей таблице, значение свойства Editable также влияет на редактируемый раскрывающийся список работает.

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'правда'		Пользователь приложения может редактировать текст или изменить

Включено	Редактирование	Визуальный результат	Функциональный результат
			выбор в редактируемом падать. Обратный звонок связано с редактируемый выпадающий выполняется, когда опция меняется.
'ложный'	'ложный'		Пользователь приложения не может изменить редактируемый выпадающий

		выделение или текст.
'правда'	'ложный'	Пользователь приложения не может изменить редактируемый выпадающий выделение или текст.
'ложный'	'правда'	Пользователь приложения не может изменить редактируемый выпадающий выделение или текст.

**Editable** - редактируемое состояние редактируемого выпадающего true (по умолчанию) | ложный

Редактируемое состояние редактируемого раскрывающегося списка, указанного как true или false.

Как показано в следующей таблице, значение свойства Enabled также влияет на редактируемый раскрывающийся список редактируемый.

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'правда'		Пользователь приложения может редактировать текст или изменить выбор в редактируемом падать. Обратный звонок связано с редактируемый выпадающий выполняется, когда опция меняется.
'ложный'	'ложный'		Пользователь приложения не может изменить редактируемый выпадающий выделение или текст.

5-58

Редактируемые раскрывающиеся свойства

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'ложный'		Пользователь приложения не может изменить редактируемый выпадающий выделение или текст.
'ложный'	'правда'		Пользователь приложения не может изменить редактируемый выпадающий выделение или текст.

**FontAngle** - Наклон символа редактируемого выпадающего текста 'нормальный' (по умолчанию) | «Наклонным»

Наклон символа редактируемого выпадающего текста, указанного как «обычный» или «курсив». настройка это свойство курсивом выбирает наклонную версию шрифта, если она доступна в приложении пользовательская система.

**FontName** - шрифт, в котором отображается редактируемый выпадающий текст 'Helvetica' (по умолчанию) | строка

Шрифт, в котором отображается редактируемое раскрывающееся значение свойства Text, заданное как строка. Если указанный шрифт недоступен для пользователя приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

**FontSize** - размер шрифта редактируемого выпадающего текста 12 (по умолчанию) | положительное число

Размер шрифта редактируемого выпадающего текста, указанный как положительное число. Единицами являются пиксели. Пример: 14

**FontWeight** - толщина текстовых символов 'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом

• «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**Расположение** - расположение редактируемого раскрывающегося списка относительно родительского [100,100] (по умолчанию) | [X, y]

5-59

Расположение редактируемого раскрывающегося списка относительно родителя, заданное как [x, y], где x и y укажите координаты от левого нижнего угла родительского контейнера до левого нижнего угла угол редактируемого раскрывающегося списка. Единицами измерения являются пиксели.

**OuterLocation** - расположение редактируемого раскрывающегося **списка** относительно родительского контейнера. [100,100] (по умолчанию) | [X, y]

Идентично местоположению для редактируемых выпадающих меню.

**OuterSize** - Размер редактируемого раскрывающегося **списка** [100, 20] (по умолчанию) | [ширина высота]

Идентичен по размеру для редактируемых выпадающих списков.

**Родитель** - Родительский контейнер редактируемого выпадающего списка объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер редактируемого раскрывающегося списка, указанный как окно приложения, панель или кнопка. групповой объект.

**Размер** - размер редактируемого раскрывающегося списка [100 20] (по умолчанию) | [ширина высота].

Размер закрытого редактируемого раскрывающегося списка, указанный как [ширина, высота]. Единицы измерения в пикселях.

**Текст** - набор доступных для редактирования опций, представленных пользователю приложения {«Вариант 1», «Вариант 2», «Вариант 3», «Вариант 4»} (по умолчанию) | 1-я-я ячейка массива строки

Набор редактируемых выпадающих меток, указанный как массив строк размером 1 на n. редактируемый раскрывающийся список отображает столько параметров, сколько есть элементов в свойстве Text стоимость.

Пример: {'Цельсия', 'Фаренгейт', 'Кельвин'}

Пример: {'1', '2', '3'}

**TextData** - набор редактируемых **выпадающих** значений {1, 2, 3, 4} (по умолчанию) | Ячейка размером 1 на n

Данные, связанные с каждым элементом свойства Text, заданные как массив ячеек размером 1 на n целых чисел, где n больше 0. Например, если вы установите свойство Text в

5-60

Редактируемые раскрывающиеся свойства

имена сотрудников, вы можете установить для свойства TextData соответствующий идентификатор сотрудника номера.

Если вы измените количество элементов в значении свойства Text, тогда TextData

свойство обновляется соответствующим образом, *если* вы ранее явно не установили TextData. если ты явно установить свойство TextData, тогда оно не обновляется автоматически, когда количество элементов в свойстве Text изменяется.

Если значение свойства TextData содержит больше элементов, чем значение свойства Text, затем MATLAB игнорирует дополнительные элементы значения свойства TextData.

Если значение свойства TextData содержит меньше элементов, чем значение свойства Text, и пользователь приложения выбирает параметр, который не имеет соответствующего элемента свойства TextData, затем MATLAB устанавливает свойство ValueData в пустой массив ячеек ({} ) и отображает предупреждение в командной строке.

Пример: {'Красный' 'Зеленый' 'Синий'}

Пример: {'10' '20' '30' '40'}

**Значение** - выбранное или введенное значение

«Вариант 1» (по умолчанию) | элемент массива свойств текста Text | строка

Выбранное или введенное значение, указанное в качестве элемента массива свойства Text или приложения введенная пользователем строка.

Если массив свойств Text пуст, MATLAB устанавливает свойство Value для введенного пользователем строка или пустой двойной массив [], когда строка не введена.

Если пользователь приложения вводит строку в редактируемом раскрывающемся списке, настройка свойства Значение становится этой строкой, значение свойства Text и TextData остается неизменным, а Значение свойства ValueData - пустой вектор.

Если массив свойств текста изменяется программно, а текущим значением является элемент массива свойств текста, то значение остается неизменным.

Если массив свойств текста изменяется программно и значение параметра больше не элемент массива свойств Text, затем MATLAB устанавливает значение Value в первый элемент массив свойств текста.

Если свойство ValueData изменяется программно, MATLAB обновляет значение свойство элемента массива свойства Text, соответствующее новому выделению.

5-61

## Страница 122

5 Ссылка на недвижимость

Если свойство Value изменяется программно на элемент в ValueData, который появляется более одного раза в ValueData, затем элемент, выбранный в приложении, соответствует первый матч.

**ValueChangedFcn** - код, выполняемый, когда пользователь приложения изменяет редактируемый раскрывающийся список вариант

[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова, выполняемая, когда пользователь приложения изменяет возможность редактирования выпадающего меню. (выбрав новую опцию или введя текст), указанный как дескриптор функции, массив ячеек содержащий дескриптор функции и дополнительные аргументы или строку.

Этот обратный вызов не выполняется, если пользователь приложения щелкает выбранную в данный момент опцию, или если значение свойства Value изменяется программно.

Пример: @function

Пример: {@function, x}

**ValueData** - данные, связанные с указанным значением  
элемент TextData | []

Данные, связанные с указанным значением, указываются как одно из следующих:

- элемент TextData

MATLAB использует isequal, чтобы определить, соответствует ли значение свойства ValueData элемент значения свойства TextData.

- пустой двойной массив ([]), если свойство TextData пустое

Если вы программно измените значение свойства ValueData, то выбор в

приложение меняется соответственно.

Если вы программно установите ValueData для элемента в TextData, который появляется больше чем один раз в TextData, то элемент, выбранный в приложении, соответствует первому совпадению.

Если вы программно измените TextData, текущий выбор в приложении не менять. Однако ValueData обновляется до новых данных выбранной опции.

Если параметр свойства Value изменяется, значение свойства ValueData обновляется до данных связано с новым значением.

**Видимый** - редактируемый выпадающий видимость  
true (по умолчанию) | ложный

5-62

Редактируемые раскрывающиеся свойства

Редактируемая видимость раскрывающегося списка, указанная как true или false. Видимая собственность определяет, отображается ли редактируемый раскрывающийся список на экране. Если видимый свойство редактируемого раскрывающегося списка имеет значение false, весь редактируемый раскрывающийся список скрыт, но вы все равно можете указать и получить доступ к его свойствам.

5-63

5 Ссылка на недвижимость

## Калибровка недвижимости

Контроль внешнего вида и поведения

Датчики - это компоненты приложения, которые представляют собой измерительный инструмент. свойства контролировать внешний вид и поведение конкретного экземпляра датчика. Модифицировать аспекты датчика, изменение значений свойств. Используйте точечную запись для обозначения конкретного объекта и свойство:

```
датчик = uigauge;
val = gauge.Value;
gauge.Value = 45;
```

**Включено** - рабочее состояние датчика

true (по умолчанию) | ложный

Рабочее состояние датчика указывается как истинное или ложное.

Если для этого свойства задано значение true, внешний вид индикатора указывает на то, что в рабочем состоянии.

Если вы установите для этого свойства значение false, внешний вид датчика будет тусклым, показывая, что датчик не работает. Тем не менее, код приложения может изменить датчик значения свойств.

**Пределы** - минимальные и максимальные значения шкалы  
[0,100] (по умолчанию) | массив

Минимальные и максимальные значения шкалы калибровки, указанные в виде числового массива из 2 элементов. Первое значение в массиве должно быть ниже, чем второе значение.

5-64

**Location** - расположение датчика относительно родительского контейнера  
[100,100] (по умолчанию) | [X, y]

Расположение датчика относительно родителя, указывается как [x, y], где x и y определяют координаты от нижнего левого угла родительского контейнера до нижнего левого угла воображаемая коробка, окружающая датчик. Единицами измерения являются пиксели.

**MajorTicks - Значения** основных отметок  
[0, 20,40,60,80,100] (по умолчанию) | массив числовых значений

Значения основных отметок, указанные в виде массива числовых значений. Любые значения массива, которые выходят за пределы значения свойства Limits не отображаются на датчике. MATLAB удаляет повторяющиеся значения массива.

Установка значения свойства MajorTicks устанавливает значение свойства MajorTicksMode равным 'руководство по эксплуатации'.

**MajorTickLabels** - основные метки

{'0', '20', '40', '60', '80', '100'} (по умолчанию) | массив ячеек строк

Метки меток основных тиков, указанные в виде массива ячеек строк. MATLAB использует каждый элемент в качестве метки для соответствующей основной отметки.

Установка значения свойства MajorTickLabels изменяет свойство MajorTickLabelsMode

значение «ручной».

Чтобы создать пустую метку для основного тика, укажите пустую строку для соответствующего Элемент значения свойства MajorTickLabels.

Если вы указали больше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB игнорирует дополнительные метки.

Если вы указали меньше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB оставляет лишние метки без меток.

**MajorTickLabelsMode** - режим создания **меток** основных **тиков**

'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания метки основного тика, указанный как один из следующих:

- 'auto' - MATLAB создает массив строк для заполнения основных меток тиков.  
MATLAB преобразует каждый элемент числового основного тика в строку, используя num2str.

5-65

## Страница 126

5 Ссылка на недвижимость

- «ручной» - вы указываете основные метки, указав MajorTickLabels стоимость имущества.

**MajorTicksMode** - режим создания **основных тиков**

'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB заполняет значение свойства MajorTicks, создавая несколько основные отметки в диапазоне, указанном свойством Limits (включительно).
- «Вручную» - Вы указываете основные тики, указав значение свойства MajorTicks.

**MinorTicks** - Незначительные значения **тиков**

[0: 4: 100] (по умолчанию) | числовой массив

Незначительные значения тиков, указанные в виде числового массива. Любые элементы MinorTicks, которые падают внешние пределы не отображаются на датчике. Если незначительный тик падает на то же значение, что и основной тик, отображается только основной тик. MATLAB удаляет дубликаты MinorTicks значения элемента. Установка значения свойства MinorTicks устанавливает MinorTicksMode Значение свойства «вручную».

Чтобы исключить незначительные отметки из индикатора, укажите пустой массив.

**MinorTicksMode** - режим создания **мелких тиков**

'auto' (по умолчанию) | 'руководство по эксплуатации'

Незначительный режим создания тиков, указанный как один из следующих:

- 'auto' - отображение незначительных тиков в позициях вокруг индикатора, определяемых вычисление разницы между самым большим основным интервалом между любыми двумя Основные последовательные основные тики и деление его на пять. Например, если основные галочки Равномерно распределенный, полученный в результате датчик имеет четыре небольших тика между каждым основным поставить галочку.

Незначительные отметки зависят только от видимых значимых отметок (то есть значимых отметок в масштабе Пределы).

MATLAB не генерирует незначительные тики для основных тиков, которые находятся за пределами масштаба.

Когда значение свойства Limits изменяется, второстепенные галочки обновляются для заполнения полного диапазон масштабирования (свойство MinorTicks обновляется соответственно).

5-66

- «manual» - вы указываете числовой массив свойства MinorTicks. MinorTicks  
Значение свойства не меняет размер или содержание само по себе.

**OuterLocation** - расположение датчика относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для датчиков.

**OuterSize** - Размер датчика  
[120 120] (по умолчанию) | [ширина высота]

Идентичен свойству размера для датчиков.

**Родитель** - Родительский контейнер датчика  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер датчика, указанный как окно приложения, панель или объект группы кнопок.

**ScaleColorLimits** - диапазон значений, определяющих начало и конец областей цветной шкалы  
[] (по умолчанию) | массив

Диапазон значений, определяющих начало и конец областей цветной шкалы, обозначаемый как n-by-2 массив числовых значений. Для каждой строки в массиве первый элемент должен быть меньше чем второй элемент.

Применяя цвета к шкале датчика, MATLAB применяет цвета, начиная с  
Первая строка в значении свойства ScaleColors. Поэтому, если две строки в ScaleColorLimits значение свойства перекрывается, тогда цвет, примененный позже, имеет приоритет.

Датчик не отображает какую-либо часть элемента значения свойства ScaleColorLimits, который выходит за пределы значения свойства Limits.

Если значения свойств ScaleColors и ScaleColorLimits имеют разные размеры, то датчик показывает только цвета, которые имеют соответствующие ограничения. Например, если свойство ScaleColors значение имеет 3 строки, но значение свойства ScaleColorLimits имеет только 2 строки, датчик отображает только первые две пары цвет / предел.

Пример: [10,20; 20,30]

**ScaleColors** - Цвета, которые появляются на шкале шкалы  
[] (по умолчанию) | RGB триплет | предварительно определенное название цвета | Ячейка размером 1 на n

Цвета, отображаемые на шкале калибровки, задаются одним из следующих:

5-67

- Массив значений RGB размером 3 на 3, где каждый элемент является допустимым значением RGB в диапазоне от 0 и 1.
- массив ячеек 1 на n, где каждый элемент является именем цвета или триплетом RGB. Клетка массив может содержать смесь RGB-триплетов и имен цветов.

MATLAB сохраняет и возвращает все имена цветов и тройки RGB в виде массива n-by-3.

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне

[0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«Y»	[1 1 0]
«Малиновый»	«M»	[1 0 1]
«Голубой»	«C»	[0 1 1]



«Красный»	'r'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«Б»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный	«К»	[0 0 0]

Каждая строка массива ScaleColors представляет цвет для *l* региона цвета шкалы.

Если вы установите свойство ScaleColors без явной установки свойства ScaleColorLimits, тогда MATLAB определяет значение свойства ScaleColorLimits, создавая п  
разнесенные, не перекрывающиеся пределы, охватывающие всю шкалу датчика. Например, если значение свойства Limits равно [0,75], а значение свойства ScaleColors равно

{'green', 'yellow', 'red'}, затем MATLAB устанавливает ScaleColorLimits на [0,25; 25,50; 50,75]

Точно так же, если вы изменяете лимиты без явной установки ScaleColorLimits, то MATLAB устанавливает значение свойства ScaleColorLimits, как описано в предыдущем параграф.

Пример: {'blue', 'green'}

Пример: {[0, .5, .2], «зеленый»}

5-68

Пример: {[0, .5, .2], [.2, .6,0]}

**ScaleDirection** - Направление масштаба  
по часовой стрелке (по умолчанию) | 'против часовой стрелки'

Направление шкалы, указанное как одно из следующего:

- «по часовой стрелке» - шкала отображается так, что значения галочки шкалы увеличиваются в по часовой стрелке.
- «против часовой стрелки» - шкала отображается так, что значения отметок шкалы увеличиваются против часовой стрелки.

**Размер** - размер датчика  
[120 120] (по умолчанию) | [ширина высота]

Размер датчика, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Значение** - расположение иглы  
0 (по умолчанию) | числовой

Расположение иглы датчика, указывается как любое числовое значение. Если значение меньше, чем минимальное значение свойства Limits, тогда стрелка указывает на местоположение непосредственно перед начало шкалы. Если значение больше максимального значения свойства Limits, затем стрелка указывает на место сразу после конца шкалы.

Изменение значения свойства Limits не влияет на настройку свойства Value.

Пример: 60

**Видимый** - видимость датчика  
true (по умолчанию) | ложный

Видимость датчика, указанная как истинная или ложная. Свойство Visible определяет, является ли Датчик отображается на экране. Если свойство Visible датчика установлено в false, весь датчик скрыт, но вы все равно можете указать и получить доступ к его свойствам.

## Knob Properties

Ручка управления внешним видом и поведением

Ручки - это компоненты приложения, представляющие ручки управления инструментом, которые пользователи могут настроить для контроля значения. Свойства контролируют внешний вид и поведение конкретного экземпляра ручки. Чтобы изменить аспекты ручки, измените значения свойств. Используйте точечную запись для ссылки на конкретный объект и свойство:

```
ручка = uiknob;  
knob.Value = 45;  
val = knob.Value;
```

**Включено** - Внешний вид и поведение  
true (по умолчанию) | ложный

Внешний вид и поведение ручки, указано как true или false.

Если вы установите для этого свойства значение true, внешний вид регулятора будет ярким, указывая на то, что приложение пользователь может переместить его.

Если для этого свойства установлено значение false, ручка будет недоступна, что означает, что пользователь приложения не может переместить его, и он не будет вызывать обратный вызов.

**Limits** - минимальное и максимальное значения регулятора  
[0,100] (по умолчанию) | массив

Минимальное и максимальное значения регулятора, заданные в виде 2-элементного числового массива.

Если вы измените лимиты так, чтобы значение регулятора было меньше, чем новый нижний предел, MATLAB устанавливает значение на новый нижний предел. Например, допустим, что пределы составляют [0,100], а Значение регулятора равно 20. Если пределы изменяются на [50,100], то MATLAB устанавливает значение регулятора до 50

Аналогично, если вы измените пределы так, что значение регулятора будет больше, чем новое верхнее значение. предел, MATLAB устанавливает значение в новый верхний предел.

**Location** - расположение кнопки, исключая отметки, относительно родительского  
[126,111] (по умолчанию) | [X, y]

Расположение кнопки, исключая отметки, относительно родителя, указано как [x, y], где

x и y задают координаты от левого нижнего угла родительского контейнера до нижний левый угол ручки. Единицами измерения являются пиксели.

**MajorTicks** - Значения основных отметок  
[0,10,20,30,40,50,60,70,80,90,100] (по умолчанию) | массив числовых значений

Значения основных отметок, указанные в виде массива числовых значений. Каждое значение должно быть равно или больше 0. Любые значения массива, выходящие за пределы ограничений, не отображаются на ручка. MATLAB удаляет дублирующиеся значения массива.

Установка значения свойства MajorTicks устанавливает значение свойства MajorTicksMode равным 'руководство по эксплуатации'.

**MajorTickLabels** - метки на каждой основной отметке  
{'0', '10', '20', '30', '40', '50', '60', '70', '80', '90', '100'} (по умолчанию) | клетка массив строк

Метки на каждой основной метке, указанной в виде массива строк. MATLAB использует каждый элемент в качестве метки для соответствующей основной отметки.

Установка MajorTickLabels изменяет значение MajorTickLabelsMode на «ручной».

Чтобы создать пустую метку для основного тика, укажите пустую строку для этого тика соответствующий элемент MajorTickLabels.

Если вы указали больше элементов MajorTickLabels, чем элементов MajorTicks, MATLAB игнорирует лишние метки.

Если вы указали меньше элементов MajorTickLabels, чем элементов MajorTicks, MATLAB оставляет лишние метки без надписей.

Пример: {'high', 'medium', 'low'}

**MajorTickLabelsMode** - режим создания меток **основных тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB создает массив строк для заполнения MajorTickLabels.  
MATLAB преобразует каждый элемент числового основного тика в строку, используя num2str.
- «manual» - вы указываете массив строк MajorTickLabels.

**MajorTicksMode** - режим создания **основных тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB заполняет MajorTicks, создавая несколько основных отметок в диапазон, указанный лимитами (включительно).
- «manual» - вы указываете массив значений MajorTicks.

**MinorTicks** - Незначительные значения отметки  
[0: 2: 100] (по умолчанию) | числовой массив

Незначительные значения отметки, указанные в виде числового массива. Любые элементы MinorTicks, которые падают внешние пределы не отображаются на датчике. Если незначительный тик падает на то же значение, что и основной тик, отображается только основной тик. MATLAB удаляет дубликаты MinorTicks значения элемента. Установка значения свойства MinorTicks устанавливает MinorTicksMode Значение свойства «вручную».

Чтобы исключить незначительные отметки из индикатора, укажите пустой массив.

**MinorTicksMode** - режим создания **мелких тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Незначительный режим создания тиков, указанный как один из следующих:

- 'auto' - MATLAB отображает небольшие тики в определенных местах вокруг ручки  
вычисляя разницу между наибольшим основным интервалом между любыми двумя Основными последовательные основные тики и деление его на пять. Например, если основные галочки равномерно распределенный, получившаяся ручка имеет четыре небольших тика между поставить галочку.

Незначительные отметки зависят только от видимых значимых отметок (то есть значимых отметок в масштабе пределы).

MATLAB не генерирует незначительные тики для основных тиков, которые находятся за пределами масштаба.

5-72

При изменении значения пределов второстепенные тики обновляются для заполнения полной шкалы диапазон (свойство MinorTicks обновляется соответственно).

- «manual» - вы указываете числовой массив MinorTicks. Значение MinorTicks не изменить размер или содержание самостоятельно.

**OuterLocation** - расположение кнопки, включая отметки, относительно родительского [100,104] (по умолчанию) | [X, y]

Расположение ручки, включая отметки, относительно родителя, указано как [x, y]. Единицы являются пикселями.

**OuterSize** - размер ручки, включая отметки [112,100] (по умолчанию) | [высота ширина]

Размер ручки, включая отметки, возвращается как [высота, ширина].

**Parent** - родительский контейнер кнопки объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер кнопки, указанный в качестве дескриптора окна приложения, панели или группы кнопок объект.

**Размер** - размер кнопки, без отметок [60,60] (по умолчанию) | [ширина высота]

Размер ручки, без отметок, задан как [ширина, высота].

**Значение** - значение ручки 0 (по умолчанию) | числовой

Значение ручки, указанное в виде числа. Числовое значение должно быть в пределах диапазона указано для лимитов.

**ValueChangedFcn** - функция обратного вызова, выполняемая, когда пользователь приложения меняет значение регулятора. [] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова для выполнения один раз, когда пользователь приложения изменяет значение ручки (перетаскивая и отпустите кнопку), указанную в качестве одного из следующих значений:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы

5-73

5 Ссылка на недвижимость

- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пример: @function

Пример: {@function, x}

**ValueChangingFcn** - функция обратного вызова, выполняемая при изменении пользователем значения приложения. [] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова, выполняемая пользователем приложения, изменяет значение ручки, указанное как единое целое. из этих значений:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Когда пользователь приложения перетаскивает ручку, MATLAB генерирует данные события и сохраняет их в Значение свойства события, к которому вы можете обратиться.

Например, чтобы отобразить значение события в командном окне MATLAB, укажите код в обратном вызове ValueChangingFcn:

```
дисп (event.Value)
```

Обратный вызов выполняется следующим образом:

- Если пользователь приложения щелкает значение ручки, обратный вызов выполняется один раз. Например, если ручка находится на 1.0, и пользователь одним щелчком мыши 1.1, обратный вызов выполняется и событие значение обновляется один раз, со значением 1,1.
- Если пользователь приложения щелкает и перетаскивает ручку на новую позицию, выполняется обратный вызов несколько раз. Например, если значение ручки равно 1,0, а пользователь щелкает, удерживает и перетаскивает до значения 1.1 обратный вызов выполняется и значение события обновляется до 1.1. Если пользователь продолжает перетаскивание до 1.2, обратный вызов выполняется снова и значение события обновляется до 1.2. Когда пользователь отпускает мышь, обратный вызов не выполняется.
- Если свойство Value регулятора изменяется программно, то обратный вызов не выполнить.

Пример: @function

Пример: {@function, x}

**Видимый** - видимость ручки  
true (по умолчанию) | ложный

5-74

Видимость ручки, указанная как true или false. Свойство Visible определяет, является ли ручка отображается на экране. Если для свойства Visible регулятора установлено значение false, вся ручка скрыта, но вы все равно можете указать и получить доступ к ее свойствам.

## Свойства ярлыка

Контроль внешнего вида этикетки

Метки - это компоненты приложения, которые используют текст для идентификации других компонентов или указывают Назначение этих других компонентов. Чтобы изменить аспекты метки, измените значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
ярлык = uilabel;  
label.Text = 'Type:';  
txt = label.Text;
```

**Включено** - Визуальное отображение метки  
true (по умолчанию) | ложный

Внешний вид этикетки, указанный как один из следующих:

- true - метка отображается нормально
- false - метка отображается тусклой

**FontAngle** - символьный уклон текста надписи  
'нормальный' (по умолчанию) | «Наклонным»

Наклон символов текста метки, указанный как «обычный» или «курсив». Установка этого свойства курсивом выбирает наклонную версию шрифта, если она доступна в системе пользователя приложения.

**FontName** - шрифт, в котором отображается метка **Text**  
'Helvetica' (по умолчанию) | строка

Шрифт, в котором отображается метка Значение свойства текста, заданное в виде строки. Если указанный шрифт недоступен для пользователя приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

**FontSize** - размер шрифта текста надписи  
12 (по умолчанию) | положительное число

Размер шрифта текста метки, указанный как положительное число. Единицами измерения являются точки.

Пример: 14

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальному весу шрифта.

**HorizontalAlignment** - Горизонтальное выравнивание текста метки  
'left' (по умолчанию) | «правильно» | 'центр'

Выравнивание текста метки, указанного как:

- 'right' - текст выравнивается по правой стороне области, указанной меткой Size свойство.
- 'left' - текст выравнивается по левой стороне области, указанной свойством метки Size.
- «центр» - текст центрируется горизонтально в области, указанной как свойство размера метки.

Выравнивание текста метки полезно, когда текст занимает несколько строк.

**Расположение** - расположение метки относительно родителя  
[100,100] (по умолчанию) | [X, y]

Расположение метки относительно родителя, указывается как [x, y], где x и y указывают координаты от нижнего левого угла родительского контейнера до нижнего левого угла этикетка. Единицами измерения являются пиксели.

**OuterLocation** - расположение метки относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для ярлыков.

5-77

## Страница 138

5 Ссылка на недвижимость

**OuterSize** - размер этикетки  
[31,15] (по умолчанию) | [ширина высота]

Идентичный по размеру для ярлыков.

**Родитель** - Ярлык контейнер  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Контейнер меток, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - Размер области отображения для этикетки  
[31,15] (по умолчанию) | [Ширина, 20].

Размер области отображения метки, указанный как [ширина, высота]. Если свойство Размер значение слишком мало для отображения значения «Текст», MATLAB обрезает текст в приложении. Единицы измерения являются пикселями.

**Текст** - текст метки  
«Метка» (по умолчанию) | строка | массив ячеек строк

Текст метки, указанный как строка или массив ячеек строк. Используйте массив строк для указать многострочную метку.

MATLAB может корректно отображать форматированный текст, например:

```
text = sprintf ( '% s \ n % s ' , 'Строка 1' , 'Строка 2' )
label = uilabel ( «Текст» , текст, «Размер» , [100,100])
```

Однако MATLAB автоматически не интерпретирует и не отображает текст, такой как этот:

```
label = uilabel ( «Текст» , «Строка 1 \ nLine2» , «Размер» , [100,150])
```

Пример: «Порог»  
Пример: {'Порог' 'Значение'}

5-78

Свойства ярлыка

**VerticalAlignment** - Вертикальное выравнивание текста  
top (по умолчанию) | «дно» | 'центр'

Вертикальное выравнивание текста метки, указанного как одно из следующего:

- «верх» - текст выравнивается по верхней части области, указанной свойством «Размер метки».
- «низ» - текст выравнивается по нижней части области, указанной размером метки свойство.
- «центр» - текст центрируется вертикально в центре области, указанной меткой Размер собственности.

Выравнивание текста метки полезно, когда текст занимает несколько строк.

**Видимый** - видимость ярлыка  
true (по умолчанию) | ложный

Видимость метки, указанная как true или false. Свойство Visible определяет, является ли метка отображается на экране. Если для свойства Visible метки установлено значение false, вся метка скрыта, но вы все равно можете указать и получить доступ к ее свойствам.

5-79

5 Ссылка на недвижимость



## Свойства лампы

Внешний вид контрольной лампы

Лампы - это компоненты приложения, которые отображают состояние с помощью цвета. Контроль свойств лампы появление конкретного экземпляра лампы. Чтобы изменить аспекты лампы, измените значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
лампа = uilamp;  
цвет = лампа. цвет;  
lamp.Color = 'red';
```

**Цвет** - цвет лампы  
[0,1,0] (зеленый) (по умолчанию) | RGB триплет | предварительно определенное название цвета

Цвет лампы, указанный как один из следующих:

- RGB триплет
- Предопределенное название цвета

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне

[0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«У»	[1 1 0]
«Малиновый»	«М»	[1 0 1]
«Голубой»	«С»	[0 1 1]
«Красный»	'r'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«Б»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный	«К»	[0 0 0]

5-80

Независимо от того, как вы указываете цвета, MATLAB сохраняет и возвращает их как RGB тройни. Например:

```
лампа = uilamp ( «Цвет» , «синий» );  
х = лампа. цвет
```

```
ANS =  
0 0 1
```

Пример: [0,1,0]

Пример: 'g'

Пример: «зеленый»

**Включено** - рабочее состояние лампы  
true (по умолчанию) | ложный

Рабочее состояние лампы указывается как истинное или ложное.

Если для этого свойства установлено значение true, лампа будет гореть ярко, указывая на то, что лампа в рабочем состоянии.

Если для этого свойства установлено значение false, лампа будет недоступна.

**Расположение** - Расположение лампы относительно родителя  
[100,100] (по умолчанию) | [X, y]

Расположение лампы относительно родителя, заданное как [x, y], где x и y указывают координаты от нижнего левого угла родительского контейнера до нижнего левого угла лампы. Единицами измерения являются пиксели.

**OuterLocation** - расположение лампы относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для ламп.

**OuterSize** - Размер лампы  
[20,20] (по умолчанию) | [ширина высота]

Идентичный по размеру для ламп.

**Родитель** - родительский контейнер лампы  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

5-81

## Страница 142

5 Ссылка на недвижимость

Родительский контейнер лампы, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - Размер лампы  
[20,20] (по умолчанию) | [ширина высота].

Размер лампы, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Видимый** - видимость лампы  
true (по умолчанию) | ложный

Видимость лампы указана как истинная или ложная. Свойство Visible определяет, является ли лампа отображается на экране. Если для свойства Visible лампы установлено значение false, вся лампа скрыта, но вы все равно можете указать и получить доступ к ее свойствам.

5-82

## Свойства линейного датчика

Контроль внешнего вида и поведения линейного датчика

Линейные градусники являются компонентами приложения, которые представляют собой измерительный прибор. Свойства контролируют внешний вид и поведение конкретного экземпляра линейного датчика. Чтобы изменить аспекты линейного датчика, измените значения свойств. Используйте точечную запись для обозначения конкретный объект и свойство:

```
датчик = uigauge («линейный»);  
val = gauge.Value;  
gauge.Value = 45;
```

### Включено - рабочее состояние линейного датчика

true (по умолчанию) | ложный

Рабочее состояние линейного датчика, указывается как true или false.

Если вы установите для этого свойства значение true, появление линейного датчика означает, что Линейный датчик работает.

Если вы установите для этого свойства значение false, линейный датчик будет затемнен, показывая, что линейный датчик не работает. Тем не менее, код приложения может изменить Значения линейных свойств.

### Пределы - минимальное и максимальное значения линейной шкалы

[0,100] (по умолчанию) | массив

Минимальные и максимальные линейные значения калибровочной шкалы, указанные в виде двухэлементного числа массив. Первое значение в массиве должно быть ниже, чем второе значение.

### Местоположение - Местоположение линейного датчика относительно родителя

[100,100] (по умолчанию) | [X, y]

Расположение линейного датчика относительно родителя, указывается как [x, y], где x и y определяют координаты от нижнего левого угла родительского контейнера до нижнего левого угла линейный датчик. Единицами измерения являются пиксели.

### MajorTicks - Значения основных отметок

[0,20,40,60,80,100] (по умолчанию) | массив числовых значений

Значения основных отметок, указанные в виде массива числовых значений. Каждое значение должно быть равно или больше 0. Любые значения массива, выходящие за пределы значения свойства Limits, не отображается на линейном индикаторе. MATLAB удаляет дублирующиеся значения массива.

Установка значения свойства MajorTicks устанавливает значение свойства MajorTicksMode равным 'руководство по эксплуатации'.

### MajorTickLabels - основные метки

{'0', '20', '40', '60', '80', '100'} (по умолчанию) | массив ячеек строк

Метки меток основных тиков, указанные в виде массива ячеек строк. MATLAB использует каждый элемент в качестве метки для соответствующей основной отметки.

Установка значения свойства MajorTickLabels изменяет свойство MajorTickLabelsMode значение «ручной».

Чтобы создать пустую метку для основного тика, укажите пустую строку для соответствующего Элемент значения свойства MajorTickLabels.

Если вы указали больше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB игнорирует дополнительные метки.

Если вы указали меньше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB оставляет лишние метки без меток.

**MajorTickLabelsMode** - режим создания **меток** основных **тиков**

'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания метки основного тика, указанный как один из следующих:

- 'auto' - MATLAB создает массив строк для заполнения основных меток тиков.  
MATLAB преобразует каждый элемент числового основного тика в строку, используя num2str.
- «ручной» - вы указываете основные метки, указав MajorTickLabels стоимость имущества.

**MajorTickLabels** - режим создания **основных тиков**

'auto' (по умолчанию) | 'руководство по эксплуатации'

5-84

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB заполняет значение свойства MajorTicks, создавая одинаково 6 размеченные основные отметки в диапазоне, указанном свойством Limits (включительно).
- «ручной» - Вы указываете основные тики, указав значение свойства MajorTicks.

**MinorTicks** - Незначительные значения **тиков**

[0: 4: 100] (по умолчанию) | числовой массив

Незначительные значения тиков, указанные в виде числового массива. Любые элементы MinorTicks, которые падают внешние пределы не отображаются на датчике. Если незначительный тик падает на то же значение, что и основной тик, отображается только основной тик. MATLAB удаляет дубликаты MinorTicks значения элемента. Установка значения свойства MinorTicks устанавливает MinorTicksMode Значение свойства «вручную».

Чтобы исключить незначительные отметки из индикатора, укажите пустой массив.

**MinorTicksMode** - режим создания **мелких тиков**

'auto' (по умолчанию) | 'руководство по эксплуатации'

Незначительный режим создания тиков, указанный как один из следующих:

- 'auto' - отображение незначительных тиков в позициях по шкале, определяемых вычисление разницы между самым большим основным интервалом между любыми двумя Основными последовательные основные тики и деление его на пять. Например, если основные галочки Равномерно распределенный, полученный в результате датчик имеет четыре небольших тика между каждым основным поставить галочку.

Незначительные отметки зависят только от видимых значимых отметок (то есть значимых отметок в масштабе Пределы).

MATLAB не генерирует незначительные тики для основных тиков, которые находятся за пределами масштаба.

Когда значение свойства Limits изменяется, второстепенные галочки обновляются для заполнения полного диапазон масштабирования (свойство MinorTicks обновляется соответственно).

- 'manual' - вы указываете числовой массив свойства MinorTicks. MinorTicks Значение свойства не меняет размер или содержание само по себе.

**Ориентация** - расположение макета линейного датчика

'горизонтальный' (по умолчанию) | «Вертикальный»

Положение макета линейного датчика, указывается как «горизонтальный» или «вертикальный».

**OuterLocation** - Расположение линейного датчика относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для линейных датчиков.

**OuterSize** - Размер линейного датчика  
[120,40] (по умолчанию) | [ширина высота]

Идентично свойству размера для линейных датчиков.

**Родитель** - Родительский контейнер линейного калибра  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер линейного размера, указанный как окно приложения, панель или группа кнопок объект.

**ScaleColorLimits** - диапазон значений, определяющих начало и конец областей цветной шкалы  
[] (по умолчанию) | массив

Диапазон значений, определяющих начало и конец областей цветной шкалы, обозначаемый как n-by-2 массив числовых значений. Для каждой строки в массиве первый элемент должен быть меньше чем второй элемент.

Применяя цвета к линейной шкале, MATLAB применяет цвета начиная с первой строки в значении свойства ScaleColors. Поэтому, если две строки в Значение свойства ScaleColorLimits перекрывается, тогда цвет, примененный позже, имеет приоритет.

Датчик не отображает какую-либо часть элемента значения свойства ScaleColorLimits, который выходит за пределы значения свойства Limits.

Если значения свойств ScaleColors и ScaleColorLimits имеют разные размеры, то датчик показывает только цвета, которые имеют соответствующие ограничения. Например, если свойство ScaleColors значение имеет 3 строки, но значение свойства ScaleColorLimits имеет только 2 строки, датчик отображает только первые две пары цвет / предел.

Пример: [10,20; 20,30]

**ScaleColors** - Цвета, которые появляются на шкале шкалы  
[] (по умолчанию) | RGB триплет | предварительно определенное название цвета | Ячейка размером 1 на n

Цвета, отображаемые на линейной шкале, задаются одним из следующих:

- Массив значений RGB размером 3 на 3, где каждый элемент является допустимым значением RGB в диапазоне от 0 и 1.

- массив ячеек 1 на n, где каждый элемент является именем цвета или триплетом RGB. Клетка массив может содержать смесь RGB-триплетов и имен цветов.

MATLAB сохраняет и возвращает все имена цветов и тройки RGB в виде массива n-by-3.

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне

[0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«Y»	[1 1 0]
«Малиновый»	«M»	[1 0 1]
«Голубой»	«C»	[0 1 1]
«Красный»	'r'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«B»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный»	«K»	[0 0 0]

Каждая строка массива ScaleColors представляет цвет для области  $i$ -й цветной шкалы.

Если вы установите свойство ScaleColors без явной установки свойства ScaleColorLimits, тогда MATLAB определяет значение свойства ScaleColorLimits, создавая п разнесенные, не перекрывающиеся пределы, охватывающие всю шкалу линейного датчика. Например, если значение свойства Limits равно [0,75], а значение свойства ScaleColors равно {'green',

'yellow', 'red'}, затем MATLAB устанавливает ScaleColorLimits на [0,25; 25,50; 50,75]

Точно так же, если вы изменяете лимиты без явной установки ScaleColorLimits, то MATLAB устанавливает значение свойства ScaleColorLimits, как описано в предыдущем параграфе.

Пример: {'blue', 'green'}

Пример: {[0, .5, .2], «зеленый»}

Пример: {[0, .5, .2], [.2, .6, 0]}

**Размер** - Размер линейного датчика  
[120,40] (по умолчанию) | [ширина высота]

5-87

## Страница 148

5 Ссылка на недвижимость

Размер линейного датчика, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Значение** - расположение иглы  
0 (по умолчанию) | числовой

Местоположение иглы линейного калибра, указывается как любое числовое значение. Если значение меньше минимальное значение свойства Limits, тогда стрелка сразу указывает на местоположение до начала шкалы. Если значение больше максимального свойства Limits значение, то стрелка указывает на место сразу после конца шкалы.

Изменение значения свойства Limits не влияет на настройку свойства Value.

Пример: 60

**Видимый** - видимость линейного датчика  
true (по умолчанию) | ложный

Видимость линейной шкалы, заданная как true или false. Видимое свойство определяет отображается ли линейный датчик на экране. Если свойство Visible линейного Датчик имеет значение false, весь линейный датчик скрыт, но вы все равно можете указать и получить доступ к его свойствам.

## Свойства списка

Контрольный список, внешний вид и поведение

Списки - это компоненты приложения, которые отображают список элементов, из которых пользователь приложения может выбрать один или несколько предметов. Как правило, списки генерируют действие, когда один или несколько элементов выбраны. Свойства управляют внешним видом и поведением конкретного экземпляра списка. Чтобы изменить аспекты списка, измените значения свойств. Используйте точечную запись для обозначения для конкретного объекта и свойства:

```
listbox = uilistbox;  
selection = listbox.Value;  
listbox.Value = 'Item 3';
```

**Включено** - рабочее состояние списка  
true (по умолчанию) | ложный

Рабочее состояние списка, указывается как true или false.

Если для этого свойства установлено значение true, пользователь приложения может сделать выбор из списка.

Если для этого свойства установлено значение false, поле со списком будет затемнено, указывая, что приложение пользователь не может сделать выбор, и окно списка не вызовет обратный вызов.

**Расположение** - расположение списка относительно родителя  
[100,100] (по умолчанию) | [X, y]

Расположение поля списка относительно родительского элемента, указанного как [x, y], где x и y указывают координаты от нижнего левого угла родительского контейнера до нижнего левого угла списка Единицами измерения являются пиксели.

**Multiselect** - если пользователь приложения может выбрать несколько элементов списка  
false (по умолчанию) | правда

Если пользователь приложения может выбрать несколько элементов списка, указывается как true или false.

**OuterLocation** - расположение списка относительно родителя  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для списков.

**OuterSize** - Размер списка  
[77,86] (по умолчанию) | [ширина высота].

Идентичен по размеру для списков.

**Родитель** - Родительский контейнер Списка  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер списка, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - Размер списка  
[77,86] (по умолчанию) | [ширина высота].

Размер списка, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Текст** - параметры списка, представленные пользователю приложения  
{'Item 1', 'Item 2', 'Item 3', 'Item 4'} (по умолчанию) | Массив строк размером 1 на n

Параметры окна списка, представленные пользователю приложения, задаются как массив строк размером 1 на n. Повторяющиеся элементы допускаются. MATLAB отображает элементы массива ячеек свойства Text сверху вниз в окне списка.

Каждая строка соответствует элементу массива ячейки свойства TextData. Например, если вы установите свойство Text для отображения имен сотрудников, вы можете установить свойство TextData в соответствующие идентификационные номера сотрудников.

Если вы укажете меньше элементов для массива свойства Text, чем для TextData  
Массив свойств, MATLAB игнорирует дополнительные элементы, указанные для TextData  
массив ячеек свойств.

Если вы укажете больше элементов для массива свойства Text, чем для TextData  
массив свойств, и пользователь приложения выбирает параметр, для которого нет соответствующих TextData  
элемент массива свойства, затем MATLAB устанавливает свойство ValueData следующим образом:  
в зависимости от значения свойства Multiselect:

- Если для свойства Multiselect установлено значение true, для свойства ValueData устанавливается значение пустой двойной массив ([ ]).

5-90

- Если для свойства Multiselect установлено значение false, то для свойства ValueData установлено значение пустой массив ячеек ({}).

пустой двойной массив, если значение свойства Multiselect равно false,

**TextData** - набор значений списка  
{1, 2, 3, 4} (по умолчанию) | Массив ячеек 1 на n | массив пустых ячеек ({}).

Набор значений списка, указанный как массив ячеек размером 1 на n или пустой массив ячеек. Повторяющиеся элементы допускаются.

Если количество элементов в свойстве Text изменяется, MATLAB автоматически обновляется массив ячеек TextData в {1, 2, 3, ... N}, где N - количество элементов в тексте массив ячеек свойств. *Однако* если вы явно установите свойство TextData, оно перестает обновляться автоматически при изменении массива свойств текста.

Пример: {'Красный', 'Зеленый', 'Синий'}

Пример: {10, 20, 30, 40}

Пример: {true, false}

**Значение** - выбор списка  
«Элемент 1» (по умолчанию) | элемент или элементы массива свойства Text

Выбор списка, указанный как элемент или элементы (если установлено свойство Multiselect в значение true) массива свойства Text.

По умолчанию это первый элемент массива свойства Text. Свойство Value  
Параметр setting должен быть элементом массива ячеек свойства Text, как определено в isequal.



Если массив ячеек свойства Text пуст, MATLAB устанавливает свойство Value в зависимости от Значение свойства Multiselect, следующим образом:

- Если для свойства Multiselect установлено значение false, MATLAB устанавливает для свойства Value значение пустой двойной массив [].
- Если для свойства Multiselect установлено значение true, MATLAB устанавливает для свойства Value значение пустой массив ячеек ({}).

Установка свойства Value обновляет выбор, отображаемый в работающем приложении. Следовательно, если значение установлено на элемент, который появляется более одного раза в свойстве Text массив ячеек, то выбранный элемент соответствует первому совпадению.

Если для свойства Multiselect установлено значение true, то выбранные элементы соответствуют элементы, которые соответствуют. Соответствующие элементы могут быть повторены в свойстве Value значение до количества раз, которое они появляются в массиве ячейки свойства Text.

Если код изменяет элементы массива ячеек свойства Text, он имеет один из следующих эффектов:

- Если выбранная опция в запущенном приложении является элементом обновленного текста массив свойств, тогда значение свойства Value останется неизменным.
- Если текущее выбранное значение в работающем приложении *не* является элементом обновленного текста массив ячеек, а затем MATLAB устанавливает свойство Value для первого элемента Text массив ячеек свойств.

Если свойство ValueData изменяется, MATLAB обновляет свойство Value до текстовой ячейки. элемент массива, соответствующий новому значению свойства ValueData.

**ValueChangedFcn** - функция обратного вызова, которая выполняется, когда пользователь приложения меняет список выборы

[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова, которая выполняется, когда пользователь приложения меняет выбор в списке, указанный как одно из следующего:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пример: @function

Пример: {@function, x}

**ValueData** - данные, связанные с выбором списка

1 (по умолчанию) | элемент TextData

Данные, связанные со списком выбора, указываются как элемент или элементы (если Для свойства Multiselect установлено значение true) массива ячеек свойства TextData.

Если массив ячеек свойства TextData пуст, допустимые значения для свойства ValueData зависит от значения свойства MultiSelect следующим образом:

- Если для свойства Multiselect установлено значение false, свойство ValueData должно быть пустым двойной массив [].

- Если для свойства Multiselect установлено значение true, свойство ValueData должно быть пустым массив ячеек ({}).

**Видимый** - видимость списка

true (по умолчанию) | ложный

Видимость списка, указанная как true или false. Свойство Visible определяет, является ли окно списка отображается на экране. Если для свойства Visible списка установлено значение false, весь список скрыт, но вы все равно можете указать и получить доступ к его свойствам.

5-93

---

Страница 154

5 Ссылка на недвижимость

## Свойства девяносто градусов

Контроль внешнего вида и поведения датчика на девяносто градусов

Датчики девяносто градусов являются компонентами приложения, которые представляют собой измерительный прибор. Свойства управляют внешним видом и поведением конкретного экземпляра девяносто градусник. Чтобы изменить аспекты датчика девяносто градусов, измените значения свойств. использование точечная нотация для обозначения конкретного объекта и свойства:

```
датчик = uigauge («девяносто градусов»);  
val = gauge.Value;  
gauge.Value = 45;
```

**Включено** - рабочее состояние датчика девяносто градусов

true (по умолчанию) | ложный

Рабочее состояние датчика девяносто градусов, заданное как true или false.

Если вы установите для этого свойства значение true, появление индикатора девяносто градусов указывает на то, что

тогдашний градусник работает.

Если вы установите для этого свойства значение false, появится индикатор девяности градусов затемняется, указывая на то, что датчик девяности градусов не работает. Тем не менее, код приложения может изменить значения свойства девяности градусов.

**Пределы** - минимальные и максимальные значения шкалы девяности градусов  
[0,100] (по умолчанию) | массив

Минимальные и максимальные значения шкалы девяности градусов, заданные как 2-элемент числовой массив. Первое значение в массиве должно быть ниже, чем второе значение.

**Расположение** - расположение датчика девяности градусов относительно родителя  
[100,100] (по умолчанию) | [X, y]

Расположение датчика девяности градусов относительно родителя, указанного как [x, y], где x и y укажите координаты от левого нижнего угла родительского контейнера до левого нижнего угла угол девяности градусов. Единицами измерения являются пиксели.

5-94

## Страница 155

Свойства девяности градусов

Если ориентация датчика такова, что закругленный край находится ближе всего к нижнему левому краю угол родительского контейнера, затем x и y определяют координаты к юго-западу угол воображаемой коробки, окружающей датчик.

**MajorTicks** - Значения основных отметок  
[0,50,100] (по умолчанию) | массив числовых значений

Значения основных отметок, указанные в виде массива числовых значений. Каждое значение должно быть равно или больше 0. Любые значения массива, выходящие за пределы значения свойства Limits, не отображаются на шкале девяности градусов. MATLAB удаляет дублирующиеся значения массива.

Установка значения свойства MajorTicks устанавливает значение свойства MajorTicksMode равным 'руководство по эксплуатации'.

**MajorTickLabels** - основные метки  
{'0', ' 50 ', ' 100 '}} (по умолчанию) | массив ячеек строк

Метки меток основных тиков, указанные в виде массива ячеек строк. MATLAB использует каждый элемент в качестве метки для соответствующей основной отметки.

Установка значения свойства MajorTickLabels изменяет свойство MajorTickLabelsMode значение «ручной».

Чтобы создать пустую метку для основного тика, укажите пустую строку для соответствующего Элемент значения свойства MajorTickLabels.

Если вы указали больше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB игнорирует дополнительные метки.

Если вы указали меньше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB оставляет лишние метки без меток.

**MajorTickLabelsMode** - режим создания меток основных тиков  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания метки основного тика, указанный как один из следующих:

- 'auto' - MATLAB создает массив строк для заполнения основных меток тиков.  
MATLAB преобразует каждый элемент числового основного тика в строку, используя num2str.
- «ручной» - вы указываете основные метки, указав MajorTickLabels стоимость имущества.

**MajorTicksMode** - режим создания основных тиков  
'auto' (по умолчанию) | 'руководство по эксплуатации'

5-95

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB заполняет значение свойства MajorTicks, создавая несколько основных отметки в диапазоне, указанном свойством Limits (включительно).
- «ручной» - Вы указываете основные тики, указав значение свойства MajorTicks.

**MinorTicks** - Незначительные значения **тиков**  
[0: 10: 100] (по умолчанию) | числовой массив

Незначительные значения тиков, указанные в виде числового массива. Любые элементы MinorTicks, которые падают внешние пределы не отображаются на датчике. Если незначительный тик падает на то же значение, что и основной тик, отображается только основной тик. MATLAB удаляет дубликаты MinorTicks значения элемента. Установка значения свойства MinorTicks устанавливает MinorTicksMode Значение свойства «вручную».

Чтобы исключить незначительные отметки из индикатора, укажите пустой массив.

**MinorTicksMode** - режим создания **мелких тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Незначительный режим создания тиков, указанный как один из следующих:

- 'auto' - отображение незначительных тиков в позициях вокруг индикатора, определяемых вычисление разницы между самым большим основным интервалом между любыми двумя Основными последовательные основные тики и деление его на пять. Например, если основные галочки Равномерно распределенный, полученный в результате датчик имеет четыре небольших тика между каждым основным поставить галочку.

Незначительные отметки зависят только от видимых значимых отметок (то есть значимых отметок в масштабе Пределы).

MATLAB не генерирует незначительные тики для основных тиков, которые находятся за пределами масштаба.

Когда значение свойства Limits изменяется, второстепенные галочки обновляются для заполнения полного диапазон масштабирования (свойство MinorTicks обновляется соответственно).

- 'manual' - вы указываете числовой массив свойства MinorTicks. MinorTicks Значение свойства не меняет размер или содержание само по себе.

**Ориентация** - Расположение макета девяносто градусов  
'северо-запад' (по умолчанию) | 'северо-восток' | «юго-запад» | «Юго-восток»

Положение макета датчика девяносто градусов, указанное как одно из следующих положений:

5-96

'северо-Запад'

'к северо-востоку'

«Юго-запад»

«Юго-восток»

**OuterLocation** - Расположение датчика девяноста градусов относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентичен местоположению для девяноста градусов.

**OuterSize** - Размер датчика девяноста градусов  
[60,60] (по умолчанию) | [ширина высота]

Идентичен свойству размера для девяноста градусных датчиков.

**Родитель** - Родительский контейнер девяноста градусов  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер с датчиком девяноста градусов, указанный как окно приложения, панель или кнопка групповой объект.

**ScaleColorLimits** - диапазон значений, определяющих начало и конец областей цветной шкалы  
[] (по умолчанию) | массив

Диапазон значений, определяющих начало и конец областей цветной шкалы, обозначаемый как n-by-2 массив числовых значений. Для каждой строки в массиве первый элемент должен быть меньше чем второй элемент.

Применяя цвета к шкале девяноста градусов, MATLAB применяет цвета начиная с первой строки в значении свойства ScaleColors. Поэтому, если две строки в Значение свойства ScaleColorLimits перекрывается, тогда цвет, примененный позже, имеет приоритет.

Датчик не отображает какую-либо часть элемента значения свойства ScaleColorLimits, который выходит за пределы значения свойства Limits.

Если значения свойств ScaleColors и ScaleColorLimits имеют разные размеры, то датчик показывает только цвета, которые имеют соответствующие ограничения. Например, если свойство ScaleColors значение имеет 3 строки, но значение свойства ScaleColorLimits имеет только 2 строки, датчик отображает только первые две пары цвет / предел.

Пример: [10,20; 20,30]

**ScaleColors** - Цвета, которые появляются на шкале шкалы  
[] (по умолчанию) | RGB триплет | предварительно определенное название цвета | Ячейка размером 1 на n

Цвета, отображаемые по шкале девяноста градусов, указаны как одно из следующего:

- Массив значений RGB размером 3 на 3, где каждый элемент является допустимым значением RGB в диапазоне от 0 и 1
- массив ячеек 1 на n, где каждый элемент является именем цвета или триплетом RGB. Клетка массив может содержать смесь RGB-триплетов и имен цветов.

MATLAB сохраняет и возвращает все имена цветов и тройки RGB в виде массива n-by-3.

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне [0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«Y»	[1 1 0]
«Малиновый»	«M»	[1 0 1]
«Голубой»	«C»	[0 1 1]
«Красный»	'r'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«B»	[0 0 1]
«Белый»	«Ш»	[1 1 1]

Длинное имя	Короткое имя	RGB Триплет
«черный»	«К»	[0 0 0]

Каждая строка массива ScaleColors представляет цвет для области *i*-й цветной шкалы.

Если вы устанавливаете свойство ScaleColors без явной установки свойства ScaleColorLimits, то MATLAB определяет значение свойства ScaleColorLimits, создавая *n* с равным интервалом, непересекающиеся пределы, которые охватывают всю шкалу датчика девяносто градусов. Например, если значение свойства Limits равно [0,75], а значение свойства ScaleColors равно {'green',

'yellow', 'red'}, затем MATLAB устанавливает ScaleColorLimits на [0,25; 25,50; 50,75]

Точно так же, если вы изменяете лимиты без явной установки ScaleColorLimits, то MATLAB устанавливает значение свойства ScaleColorLimits, как описано в предыдущем параграфе.

Пример: {'blue', 'green'}

Пример: {[0, .5, .2], «зеленый»}

Пример: {[0, .5, .2], [.2, .6,0]}

**ScaleDirection** - Направление масштаба  
по часовой стрелке (по умолчанию) | 'против часовой стрелки'

Направление шкалы, указанное как одно из следующего:

- «по часовой стрелке» - шкала отображается так, что значения галочки шкалы увеличиваются в по часовой стрелке.
- «против часовой стрелки» - шкала отображается так, что значения отметок шкалы увеличиваются против часовой стрелки.

**Размер** - Размер датчика девяносто градусов  
[60,60] (по умолчанию) | [ширина высота].

Размер датчика девяносто градусов, указанный как [ширина, высота]. Ширина и высота Датчик определяется по длине двух прямых краев датчика, независимо от ориентации. Единицами измерения являются пиксели.

**Значение** - расположение иглы  
0 (по умолчанию) | числовой

Расположение иглы девяносто градусного датчика, указанное в виде любого числового значения. Если значение меньше минимального значения свойства Limits, тогда стрелка указывает на местоположение

непосредственно перед началом шкалы. Если значение больше максимального Ограничивает значение свойства, затем стрелка указывает на местоположение сразу после окончания масштаб.

Изменение значения свойства Limits не влияет на настройку свойства Value.

Пример: 60

**Видимый** - видимость девяносто градусов  
true (по умолчанию) | ложный

Видимость по девяносто градусам, указанная как истинная или ложная. Видимая собственность определяет, отображается ли на экране индикатор девяносто градусов. Если видимый свойство датчика девяносто градусов установлено в false, весь датчик девяносто градусов

5-100

## Числовые свойства поля редактирования

Управление внешним видом и поведением числового поля редактирования

Поля редактирования чисел - это компоненты приложения, которые позволяют пользователям приложения вводить числовые значения. Как правило, числовые поля редактирования генерируют действие, когда пользователь приложения изменяет значение. Свойства управляют внешним видом и поведением конкретного экземпляра числового редактирования поле. Чтобы изменить аспекты числового поля редактирования, измените значения свойств. Используйте точечную запись для ссылки на конкретный объект и свойство:

```
число поле = uieditfield («числовой»);  
val = numberfield.Value;  
numberfield.Value = 20;
```

**Включено** - рабочее состояние поля редактирования чисел  
true (по умолчанию) | ложный

Рабочее состояние числового поля редактирования, заданное как true или false.

Если для этого свойства задано значение true, числовое поле редактирования принимает ввод.

Если вы установите для этого свойства значение false, появится внешний вид числового поля редактирования. затемненный, указывая, что он не принимает ввод, и числовое поле редактирования не сработает обратный звонок.

Как показано в следующей таблице, значение свойства Editable также влияет на числовое поле редактирования является действующим.

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'правда'		Пользователь приложения может обновить поле редактирования. обратный вызов связан с полем редактирования будет выполнить, когда текст изменения.

'ложный'	'ложный'	Пользователь приложения не может обновить поле редактирования.
----------	----------	--

5-101

Страница 162

5 Ссылка на недвижимость

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'ложный'		Пользователь приложения не может обновить поле редактирования.
'ложный'	'правда'		Пользователь приложения не может обновить поле редактирования.

**Editable** - редактируется ли числовое поле редактирования  
true (по умолчанию) | ложный

Можно ли редактировать числовое поле редактирования, указав значение true или false.

Как показано в следующей таблице, значение свойства Enabled также влияет на числовое поле редактирования доступно для редактирования.

Включено	Редактирование	Визуальный результат	Функциональный результат
'правда'	'правда'		Пользователь приложения может обновить поле редактирования. обратный вызов связан с полем редактирования будет выполнен, когда текст изменения.
'ложный'	'ложный'		Пользователь приложения не может обновить поле редактирования.
'правда'	'ложный'		Пользователь приложения не может обновить поле редактирования.
'ложный'	'правда'		Пользователь приложения не может обновить поле редактирования.

**HorizontalAlignment** - Горизонтальное выравнивание чисел в числовом поле редактирования  
'right' (по умолчанию) | 'оставил'

Горизонтальное выравнивание чисел в числовом поле редактирования, заданное как:

- «вправо» - числа располагаются справа от поля ввода чисел.
- «влево» - цифры выравниваются с левой стороны поля ввода чисел.

5-102

Страница 163

Числовые свойства поля редактирования

**Пределы** - минимальные и максимальные числовые значения поля редактирования  
[0,100] (по умолчанию) | массив

Минимальные и максимальные значения, указанные в виде 2-элементного числового массива. Первое значение должно быть ниже, чем второе значение. Установите элементы массива в -Inf или Inf, чтобы указать по



минимум или нет максимум, соответственно.

Пример: [-Inf, 200]

Пример: [-100, Inf]

Пример: [-100,200]

**Местоположение** - Расположение числового поля редактирования относительно родителя.  
[100,100] (по умолчанию) | [X, y]

Расположение числового поля редактирования относительно родителя, указывается как [x, y], где x и y укажите координаты от левого нижнего угла родительского контейнера до левого нижнего угла угол числового поля редактирования. Единицами измерения являются пиксели.

**LowerLimitInclusivity** - может ли **значение** свойства **Value** быть равно **Limit**  
свойство ниже стоимости  
false (по умолчанию) | правда

Может ли значение свойства Value быть равно нижнему значению свойства Limits, указано как одно из следующих:

- true - значение свойства Value должно быть больше или равно нижнему пределу
- false - значение свойства должно быть больше нижнего предела

**OuterLocation** - расположение числового поля редактирования относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично расположению для числовых полей редактирования.

**OuterSize** - Размер числового поля редактирования, включая **текстовое** значение  
[100,20] (по умолчанию) | [ширина высота]

Идентично размеру для числовых полей редактирования.

5-103

## Страница 164

5 Ссылка на недвижимость

**Родитель** - Родительский контейнер числового поля редактирования  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер числового поля редактирования, указанный как окно приложения, панель или группа кнопок объект.

**RoundFractionalValues** - изменяет ли MATLAB дробное значение, введенное приложением  
пользователь на весь номер  
false (по умолчанию) | правда

Изменяет ли MATLAB дробное значение, введенное пользователем приложения, на целое номер, указанный как одно из следующего:

- true - MATLAB округляет значение, если оно приводит к допустимому значению, а затем выполняет обратный вызов ValueChangedFcn. Если полученное значение находится за пределами нижнего или верхнего ограничивает, затем MATLAB возвращает значение к предыдущему значению и не выполняет обратный вызов ValueChangedFcn.
- false - MATLAB не изменяет заданное пользователем дробное значение на целое номер.

Если значение свойства RoundFractionalValues изменяется от ложного к истинному во время выполнения приложение, то MATLAB применяет эти правила:

- Если при округлении существующего значения выдается целое число, которое находится внутри предельного диапазона определяется свойством Limits, затем MATLAB округляет существующее значение.
- Если при округлении существующего значения получается целое число, которое меньше нижнего предела, то MATLAB округляет существующее значение.
- Если при округлении существующего значения получается целое число, превышающее верхний предел, затем MATLAB округляет существующее значение.
- Если ограничения настроены так, что здесь нет действительного целого числа в диапазоне, то MATLAB возвращает RoundFractionalValues в значение false и отображает сообщение об ошибке.

**Размер** - Размер числового поля редактирования  
[100,20] (по умолчанию) | [ширина, высота]

Размер числового поля редактирования, указанный как [ширина, высота]. Единицы измерения являются пикселями.

**UpperLimitInclusivity** - может ли **значение** свойства **Value** быть равно **Limit**  
верхнее значение свойства  
false (по умолчанию) | правда

5-104

Может ли Value быть равно верхнему значению свойства Limits, указанному как одно из следующий:

- true - значение свойства Value должно быть равно или меньше верхнего предела
- false - значение свойства должно быть меньше верхнего предела

Например, если вы хотите, чтобы числовой ввод находился в диапазоне от 0 до 1, исключая 0 и 1, установите значение свойства Limit до [0,1], свойство UpperLimitInclusivity в false и Свойство LowerLimitInclusivity имеет значение false.

**Значение** - значение в числовом поле редактирования  
0 (по умолчанию) | число с двойной точностью

Значение в числовом поле редактирования, указанное как число с двойной точностью.

Когда пользователь приложения вводит или изменяет значение в числовом поле редактирования, это строка. когда пользователь нажимает **клавишу Enter** или меняет фокус, MATLAB преобразует введенный пользователем строка в число с двойной точностью.

MATLAB отклоняет значение, если:

- Он не может преобразовать строку в скалярное число.
- Значение - NaN, пустое или комплексное число.
- Значение является математическим выражением, например, 1 + 2.
- Значение меньше нижнего предела свойства Limit или больше верхнего предела.

Когда MATLAB отклоняет введенное пользователем значение, появляется всплывающая подсказка с описанием значения требования. Текст поля редактирования немедленно возвращается к предыдущему числовому полю редактирования стоимость. В этом случае значение ValueChangedFcn не запускается.

Пример: 10

**ValueChangedFcn** - код, выполняемый, когда пользователь приложения меняет числовое значение поля редактирования  
дескриптор функции (по умолчанию) | ручка функции | массив ячеек | строка

Код, выполняемый, когда пользователь приложения изменяет числовое значение поля редактирования, указанное как одно из эти значения:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

5-105

5 Ссылка на недвижимость

Эта функция выполняется всякий раз, когда пользователь приложения изменяет значение в числовом редакторе поле и меняет фокус или нажимает **клавишу Enter** . Эта функция не выполняется, если

числовое значение поля редактирования изменяется программно.  
Пример: @ имя-функции

Пример: {@ имя-функции, аргумент}

**ValueDisplayFormat** - формат цифры в поле редактирования  
"% 11.4g" (по умолчанию) | строка формата

Формат цифры в числовом поле редактирования, заданный в виде строки формата.

MATLAB использует num2str (NumericDisplayFormat, Value) для преобразования поля редактирования

Значение свойства в строку и отображает его.

Вы можете смешивать текст со строками формата. Например:

```
numfield = uieditfield ( 'numeric' , 'ValueDisplayFormat' , '% .0f MS / s' );
```

Результирующее числовое поле редактирования выглядит следующим образом:

Когда пользователь щелкает поле числового редактирования, в поле отображается значение без строка.

Полный список поддерживаемых форматных строк смотрите в num2str в MATLAB. документация.

Пример: "% .0f MS / s"

**Видимый** - Видимость числового поля редактирования  
true (по умолчанию) | ложный

Видимость числового поля редактирования, заданная как true или false. Видимая собственность определяет, будет ли числовое поле редактирования отображаться на экране. Если видимый свойство числового поля редактирования имеет значение false, все числовое поле редактирования скрыто, но вы все равно можете указать и получить доступ к его свойствам.

5-106

## Свойства панели

Внешний вид панели управления

Панели - это компоненты приложения, которые содержат другие компоненты. Как правило, вы используете их для организовать компоненты в вашем приложении. Свойства контролируют внешний вид и поведение конкретный экземпляр панели. Чтобы изменить аспекты панели, измените значения свойств. использование точечная нотация для обозначения конкретного объекта и свойства:

```
window = appwindow;  
panel = matlab.ui.control.Panel («Родитель», окно);  
title = panel.Title;  
panel.Title = 'Параметры отображения';
```

**BorderVisibility** - отображается ли граница панели  
true (по умолчанию) | ложный

Отображается ли граница панели, указанная как true или false.

**Дети** - Дети панели  
пустой массив GraphicsPlaceholder (по умолчанию) | 1-D массив компонентных объектов

Дочерние элементы панели, возвращаемые как пустой объект GraphicsPlaceholder или как 1-D массив составные объекты. Дочерними элементами панели может быть любой компонент, в том числе другой панель.

5-107

Вы не можете добавлять или удалять дочерние компоненты, используя свойство Children панель. Используйте это свойство для просмотра списка детей или для изменения порядка детей. порядок дочерних элементов в этом массиве отражает порядок «от начала к концу» (порядок размещения) компоненты на экране.

Чтобы добавить дочерний элемент в этот список, задайте для этой панели свойство Parent дочернего компонента.

**Включено** - рабочее состояние панели  
true (по умолчанию) | ложный

Внешний вид панели, указанный как

- true - Panel и ее дочерние компоненты (если вы не отключили дочерний компонент в оно) появиться в рабочем состоянии.
- false - Panel и ее дочерние компоненты отображаются затемненными, и вы не можете манипулировать любые компоненты внутри него.

Примечание Изменения Enabled свойства панели имеет *не* изменять значение свойств Включенные свойства его дочерних компонентов, даже если отключение панели вызывает дети будут инвалидами.

**FontAngle** - символьный наклон заголовка панели  
'нормальный' (по умолчанию) | «Наклонным»

Наклон символа в заголовке панели, указанный как «обычный» или «курсив». Установка этого свойство курсивом выбирает наклонную версию шрифта, если она доступна пользователю приложения система.

**FontName** - шрифт, в котором отображается заголовок панели  
'Helvetica' (по умолчанию) | строка

Шрифт для отображения значения свойства заголовка панели, заданного в виде строки. Если указанный шрифт недоступен пользователю приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

Пример: «Ариал»

**FontSize** - размер шрифта заголовка панели  
12 (по умолчанию) | положительное число

5-108

Размер шрифта заголовка панели, указанный как положительное число. Единицами измерения являются точки.

Пример: 14

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальному весу шрифта.

**Расположение** - расположение панели относительно родителя  
[100,100] (по умолчанию) | [X, y]

Расположение панели относительно родителя, указывается как [x, y], где x и y определяют координаты от нижнего левого угла родительского контейнера до нижнего левого угла панель. Единицами измерения являются пиксели.

**OuterLocation** - расположение панели относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для панелей.

**OuterSize** - размер панели, включая строку заголовка  
[260,221] (по умолчанию) | [ширина высота]

Размер панели, включая строку заголовка, возвращается как [ширина, высота]. Единицы измерения в пикселях.

**Родитель** - родительский контейнер панели  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер панели, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - размер панели, исключая строку заголовка  
[260 200] (по умолчанию) | [ширина высота]

Размер панели, исключая строку заголовка, указывается как [ширина, высота]. Единицы измерения в пикселях.

5-109

**Заголовок** - Заголовок панели  
«Панель» (по умолчанию) | строка

Заголовок панели, указанный в виде строки.

**Видимый** - Видимость панели  
'on' (по умолчанию) | «Выключено»

Видимость ручки панели, указанная как «вкл» или «выкл». Видимое свойство определяет отображается ли панель на экране. Если для свойства Visible панели установлено значение 'off', вся панель скрыта, но вы все равно можете указать и получить доступ к ее свойствам.

Примечание Изменения Visible свойства панели имеет *не* изменять значения Видимого свойства его дочерних компонентов, даже если скрытие панели вызывает компоненты быть скрытым

5-110

---

Страница 171

Свойства радиокнопок

## Свойства радиокнопок

Контроль появления радио кнопки

Радиокнопки - это компоненты приложения, которые отображаются в группе кнопок как набор эксклюзивные варианты. Как правило, группа кнопок, к которой принадлежит переключатель генерирует действие при изменении выбора. Свойства контролируют внешний вид и поведение конкретного экземпляра переключателя. Чтобы изменить аспекты переключателя, изменить значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
radiobutton = uiradiobutton;  
txt = radiobutton.Text;  
radiobutton.Text = 'One';
```

Примечание. Чтобы указать, как ваше приложение должно реагировать, когда пользователь меняет переключатель выберите код обратного вызова `ValueChangedFcn` для группы кнопок, к которой относится радио кнопки являются родителями.

**Включено** - рабочее состояние переключателя

true (по умолчанию) | ложный

Рабочее состояние переключателя, заданное как true или false.

Если для этого свойства установлено значение true, появление переключателя означает, что приложение пользователь может выбрать его.

Если вы установите для этого свойства значение false, радио-кнопка будет выглядеть тусклой, указывает на то, что пользователь приложения не может выбрать его.

**FontAngle** - символьный наклон текста переключателя

'нормальный' (по умолчанию) | «Наклонным»

Наклон символа радиокнопки, указанный как «обычный» или «курсив». настройка это свойство курсивом выбирает наклонную версию шрифта, если она доступна в приложении пользовательская система.

**FontName** - шрифт для отображения текста переключателя

'Helvetica' (по умолчанию) | строка

5-111

## Страница 172

5 Ссылка на недвижимость

Шрифт для отображения переключателя Значение свойства текста, указанное в виде строки. Если указанный шрифт недоступен для пользователя приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

Пример: «Ариал»

**FontSize** - размер шрифта текста переключателя  
12 (по умолчанию) | положительное число

Размер шрифта текста переключателя, указанный как положительное число. Единицы измерения являются точками.

Пример: 14

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальному весу шрифта.

**Расположение** - расположение переключателя относительно группы кнопок  
[10,10] (по умолчанию) | [X, y]

Расположение переключателя относительно контейнера группы родительских кнопок, указанного как

[x, y], где x и y определяют координаты из нижнего левого угла кнопки группа контейнеров в нижнем левом углу переключателя. Единицы измерения являются пикселями.

**OuterLocation** - расположение переключателя относительно родительского контейнера.  
[10,10] (по умолчанию) | [X, y]

Идентично местоположению для кнопок радио.

**OuterSize** - размер переключателя, включая текст  
[123,106] (по умолчанию) | [ширина высота]

Идентичен по размеру для радио кнопок.

5-112

## Страница 173

Свойства радиокнопок

**Родитель** - контейнер группы родительских кнопок переключателя  
объект группы кнопок

Контейнер группы родительских кнопок переключателя, указанный как объект группы кнопок. За подробности см. в группе кнопок.

**Размер** - размер переключателя, включая текст  
[123,85] (по умолчанию) | [ширина высота]

Размер переключателя, включая значение свойства Text, задается как [ширина, высота]. Единицами измерения являются пиксели.

**Текст** - текст радио кнопки  
«Радиокнопка» (по умолчанию) | строка

Текст переключателя, указанный в виде строки.

Пример: «Устойчивое состояние»

**Значение** - состояние выбора переключателя  
правда | ложный

Состояние выбора радиокнопки, заданное как true или false.

В пределах данной группы переключателей можно выбрать одну и только одну кнопку время.

Когда свойство Value для переключателя меняется на true, свойство Value для ранее выбранный переключатель, связанный с той же группой кнопок, становится ложным. В Кроме того, значение свойства SelectedObject группы кнопок обновляется в соответствии с выбранный переключатель

Если вы программно измените свойство Value для переключателя на false, MATLAB устанавливает первый переключатель в группе кнопок на true. Если первое радио кнопка, для которой вы устанавливаете свойство Value для переключателя в значение false, затем MATLAB устанавливает второй переключатель в группе кнопок на true.

**Видимый** - видимость переключателя  
true (по умолчанию) | ложный

Радиокнопка видимости, указанная как true или false. Видимое свойство определяет отображается ли переключатель на экране. Если Видимое свойство радио

5-113

---

## Страница 174

5 Ссылка на недвижимость

Кнопка имеет значение false, весь переключатель скрыт, но вы все равно можете указать и получить доступ к его свойствам.



## Свойства клавишного переключателя

Контроль внешнего вида и поведения клавишного переключателя

Клавишные переключатели - это компоненты приложения, которые указывают на логическое состояние. Как правило, рокер переключатели генерируют действие при изменении состояния. Свойства контролируют внешний вид и поведение конкретного экземпляра тумблера. Чтобы изменить аспекты рокер переключить, изменить значения свойств. Используйте точечную запись для обозначения конкретного объекта и свойство:

```
rswitch = uiswitch ('рокер');
val = rswitch.Value;
rswitch.Value = true;
```

**Включено** - рабочее состояние клавишного переключателя  
true (по умолчанию) | ложный

Рабочее состояние кулисного переключателя указано как true или false.

Если вы установите для этого свойства значение true, пользователь приложения сможет сдвинуть тумблер.

Если вы установите для этого свойства значение false, тумблер будет недоступен, указывая на то, что Пользователь приложения не может сдвинуть тумблер и не вызовет обратный вызов.

**Location** - расположение клавишного переключателя, исключая метки состояния, относительно родительского [100,121] (по умолчанию) | [X, y]

Расположение клавишного переключателя, исключая метки состояния, относительно родительского элемента, указанного как [x, y], где x и y определяют координаты из нижнего левого угла родительского контейнера в нижний левый угол кулисного переключателя. Единицами измерения являются пиксели.

**Ориентация** - Направление, в котором отображается тумблер  
'вертикальный' (по умолчанию) | «Горизонтальные»

Направление, в котором отображается тумблер, указано как «горизонтальное» или «Вертикальный».

**OuterLocation** - расположение **клавишного** переключателя, включая метки состояния, относительно родительского [100,100] (по умолчанию) | [X, y]

Расположение клавишного переключателя, включая метки состояния, относительно родителя, возвращается как [x, y], где x и y задают координаты из левого нижнего угла родителя  
Контейнер до нижнего левого угла кулисного переключателя, включая метки состояния. Единицы измерения в пикселях.

**OuterSize** - размер переключателя, включая метки состояния  
[20,87] (по умолчанию) | [ширина высота]

Размер клавишного переключателя, включая метки состояния, возвращается как [ширина, высота].

**Родитель** - Родительский контейнер клавишного переключателя  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер клавишного переключателя, указанный как окно приложения, панель или группа кнопок объект.

**Размер** - размер клавишного переключателя, без учета государственных меток  
[20,45] | [ширина высота]

Размер клавишного переключателя, исключая метки состояния, определяется как [ширина, высота]. Единицы измерения являются пикселями.

**Текст** - метки состояния Rocker Switch  
{ 'Off', 'On' } (по умолчанию) | Строки массива ячеек n-на-2

Метки состояния Rocker Switch, заданные в виде массива ячеек размером n × 2.

Когда свойство Orientation клавишного переключателя установлено на «вертикальный», первый элемент из массива появляется в нижней части кулисного переключателя. Когда свойство Ориентация установлен на «горизонтальный», первый элемент массива появляется слева от коромысла переключатель.

**Значение** - состояние клавишного переключателя  
false (по умолчанию) | правда

Состояние кулисного переключателя, заданное одним из следующих значений:

- ложный

5-116

## Страница 177

Свойства клавишного переключателя

Клавишный переключатель направлен вниз, когда свойство Orientation установлено на «вертикальное». Он указывает влево, когда для свойства Orientation установлено значение «горизонтальный».

- правда

Клавишный переключатель направлен вверх, когда свойство Orientation установлено в вертикальное положение. Это указывает вправо, когда свойство Orientation установлено на «горизонтальный».

**ValueChangedFcn** - код для выполнения, когда пользователь приложения **щелкает тумблером**  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Код, выполняемый, когда пользователь приложения переключает элемент управления с помощью клавишного переключателя, указанного как один из следующий:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пользователь приложения может изменить состояние клавишного переключателя, нажав и отпустив мышью в любом месте на тумблере (включая метки состояния) или нажав на тумблер, перетаскивая, а затем отпуская кнопку мыши, пока он находится на тумблере переключатель.

Пример: @function

Пример: {@function, x}

**Видимый** - видимость переключателя  
true (по умолчанию) | ложный

Переключатель видимости, указанный как true или false. Видимое свойство определяет отображается ли тумблер на экране. Если видимое свойство ровера переключатель установлен в значение false, весь клавишный переключатель скрыт, но вы все равно можете указать и получить доступ к его свойствам.

## Свойства полукруглого датчика

Контроль внешнего вида полукруглого датчика

Полукруглые датчики являются компонентами приложения, которые представляют собой измерительный прибор. Свойства контролируют внешний вид и поведение конкретного экземпляра полукруглой формы. калибровать. Чтобы изменить аспекты полукруглого датчика, измените значения свойств. Используйте точку запись для обозначения конкретного объекта и свойства:

```
датчик = uigauge («полукруглый»);  
val = gauge.Value;  
gauge.Value = 45;
```

**Включено** - рабочее состояние полукруглого датчика

true (по умолчанию) | ложный

Рабочее состояние полукруглого датчика, указывается как истинное или ложное.

Если вы установите для этого свойства значение true, появление полукруглого датчика означает, что Эти полукруглые датчики работоспособны.

Если вы установите для этого свойства значение false, появится вид полукруглого датчика затемненный, указывая на то, что полукруглый датчик не работает. Тем не менее, код приложения может изменить значения свойства полукруглого датчика.

**Пределы** - минимальные и максимальные значения полукруглой шкалы  
[0,100] (по умолчанию) | массив

Минимальные и максимальные полукруглые калибровочные значения, указанные как 2-элемент числовой массив. Первое значение в массиве должно быть ниже, чем второе значение.

**Location** - расположение полукруглого датчика относительно родительского контейнера  
[100,100] (по умолчанию) | [X, Y]

Расположение полукруглого датчика относительно родителя, заданное как [x, y], где x и y укажите координаты от левого нижнего угла родительского контейнера до левого нижнего угла угол полукруглого датчика. Единицами измерения являются пиксели.

Если ориентация датчика такова, что закругленный край находится ближе всего к нижнему левому краю угол родительского контейнера, затем x и y определяют координаты к юго-западу угол воображаемой коробки, окружающей датчик.

**MajorTicks** - Значения основных отметок

[0,10,20,30,40,50,60,70,80,90,100] (по умолчанию) | массив числовых значений  
Значения основных отметок, указанные в виде массива числовых значений. Каждое значение должно быть равно или больше 0. Любые значения массива, выходящие за пределы значения свойства Limits, не отображаются на полукруглом датчике. MATLAB удаляет дублирующиеся значения массива.

Установка значения свойства MajorTicks устанавливает значение свойства MajorTicksMode равным 'руководство по эксплуатации'.

**MajorTickLabels** - основные метки  
{'0', '10', '20', '30', '40', '50', '60', '70', '80', '90', '100'} (по умолчанию) | клетка массив строк

Метки меток основных тиков, указанные в виде массива ячеек строк. MATLAB использует каждый элемент в качестве метки для соответствующей основной отметки.

Установка значения свойства MajorTickLabels изменяет свойство MajorTickLabelsMode значение «ручной».

Чтобы создать пустую метку для основного тика, укажите пустую строку для соответствующего Элемент значения свойства MajorTickLabels.

Если вы указали больше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB игнорирует дополнительные метки.

Если вы указали меньше элементов значения свойства MajorTickLabels, чем свойство MajorTicks Значения элементов, MATLAB оставляет лишние метки без меток.

**MajorTicksMode** - режим создания **основных тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB заполняет значение свойства MajorTicks, создавая несколько основные отметки в диапазоне, указанном свойством Limits (включительно).
- «ручной» - Вы указываете основные тики, указав значение свойства MajorTicks.

5-119

**MajorTickLabelsMode** - режим создания **меток** основных **тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания метки основного тика, указанный как один из следующих:

- 'auto' - MATLAB создает массив строк для заполнения основных меток тиков.  
MATLAB преобразует каждый элемент числового основного тика в строку, используя num2str.
- «ручной» - вы указываете основные метки, указав MajorTickLabels стоимость имущества.

**MinorTicks** - Незначительные значения **тиков**  
[0: 4: 100] (по умолчанию) | числовой массив

Незначительные значения тиков, указанные в виде числового массива. Любые элементы MinorTicks, которые падают внешние пределы не отображаются на датчике. Если незначительный тик падает на то же значение, что и основной тик, отображается только основной тик. MATLAB удаляет дубликаты MinorTicks значения элемента. Установка значения свойства MinorTicks устанавливает MinorTicksMode Значение свойства «вручную».

Чтобы исключить незначительные отметки из индикатора, укажите пустой массив.

**MinorTicksMode** - режим создания **мелких тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Незначительный режим создания тиков, указанный как один из следующих:

- 'auto' - отображение незначительных тиков в позициях вокруг индикатора, определяемых вычисление разницы между самым большим основным интервалом между любыми двумя Основными последовательные основные тики и деление его на пять. Например, если основные галочки Равномерно распределенный, полученный в результате датчик имеет четыре небольших тика между каждым основным

поставить галочку.

Незначительные отметки зависят только от видимых значимых отметок (то есть значимых отметок в масштабе Пределы).

MATLAB не генерирует незначительные тики для основных тиков, которые находятся за пределами масштаба.

Когда значение свойства Limits изменяется, второстепенные галочки обновляются для заполнения полного диапазон масштабирования (свойство MinorTicks обновляется соответственно).

- 'manual' - вы указываете числовой массив свойства MinorTicks. MinorTicks  
Значение свойства не меняет размер или содержание само по себе.

5-120

Свойства полукруглого датчика

**Ориентация** - расположение макета полукруглого датчика  
'север' (по умолчанию) | 'юг' | 'восток' | «Запад»

Положение раскладки полукруглого манометра, указанного как одно из следующего:

'к северу'

«Юг»

«Восток»

«Запад»

**OuterLocation** - расположение полукруглого датчика относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для полукруглых датчиков.

**OuterSize** - Размер полукруглого датчика  
[120,65] (по умолчанию) | [ширина высота]

Идентично свойству размера для полукруглых датчиков.

5-121

5 Ссылка на недвижимость

**Родитель** - Родительский контейнер полукруглой формы  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер полукруглого датчика, указанный как окно приложения, панель или кнопка групповой объект.

**ScaleColorLimits** - диапазон значений, определяющих начало и конец областей цветной шкалы  
[] (по умолчанию) | массив

Диапазон значений, определяющих начало и конец областей цветной шкалы, обозначаемый как n-by-2 массив числовых значений. Для каждой строки в массиве первый элемент должен быть меньше чем второй элемент.

Применяя цвета к шкале полукруглого датчика, MATLAB применяет цвета начиная с первой строки в значении свойства ScaleColors. Поэтому, если две строки в Значение свойства ScaleColorLimits перекрывается, тогда цвет, примененный позже, имеет приоритет.

Датчик не отображает какую-либо часть элемента значения свойства ScaleColorLimits, который выходит за пределы значения свойства Limits.

Если значения свойств ScaleColors и ScaleColorLimits имеют разные размеры, то датчик показывает только цвета, которые имеют соответствующие ограничения. Например, если свойство ScaleColors значение имеет 3 строки, но значение свойства ScaleColorLimits имеет только 2 строки, датчик отображает только первые две пары цвет / предел.

Пример: [10,20; 20,30]

**ScaleColors** - Цвета, которые появляются на шкале шкалы  
[] (по умолчанию) | RGB триплет | предварительно определенное название цвета | Ячейка размером 1 на n

Цвета, которые отображаются на полукруглой шкале, указаны как одно из следующего:

- Массив значений RGB размером 3 на 3, где каждый элемент является допустимым значением RGB в диапазоне от 0 и 1
- массив ячеек 1 на n, где каждый элемент является именем цвета или триплетом RGB. Клетка массив может содержать смесь RGB-триплетов и имен цветов.

MATLAB сохраняет и возвращает все имена цветов и тройки RGB в виде массива n-by-3.

RGB-триплет представляет собой трехэлементный вектор-строку, элементы которого задают интенсивности красный, зеленый и синий компоненты цвета. Интенсивности должны быть в диапазоне

5-122

[0,1], например, [0,4 0,6 0,7]. В этой таблице перечислены триплетные значения RGB, которые имеют эквивалентные цветные строки.

Длинное имя	Короткое имя	RGB Триплет
«Желтый»	«У»	[1 1 0]
«Малиновый»	«М»	[1 0 1]
«Голубой»	«С»	[0 1 1]
«Красный»	'р'	[1 0 0]
«Зеленый»	'грамм'	[0 1 0]
«Синий»	«Б»	[0 0 1]
«Белый»	«Ш»	[1 1 1]
«черный	«К»	[0 0 0]

Каждая строка массива ScaleColors представляет цвет для области i-й цветной шкалы.

Если вы устанавливаете свойство ScaleColors без явной установки свойства ScaleColorLimits, то MATLAB определяет значение свойства ScaleColorLimits, создавая n с равным интервалом, непересекающиеся пределы, охватывающие всю шкалу полукруглого датчика. Например, если значение свойства Limits равно [0,75], а значение свойства ScaleColors равно {'green', 'yellow', 'red'}, затем MATLAB устанавливает ScaleColorLimits на [0,25; 25,50; 50,75]

Точно так же, если вы изменяете лимиты без явной установки ScaleColorLimits, то

MATLAB устанавливает значение свойства ScaleColorLimits, как описано в предыдущем параграфе.

Пример: {'blue', 'green'}

Пример: {[0, .5, .2], «зеленый»}

Пример: {[0, .5, .2], [.2, .6, 0]}

**ScaleDirection** - Направление масштаба

по часовой стрелке (по умолчанию) | 'против часовой стрелки'

Направление шкалы, указанное как одно из следующего:

- «по часовой стрелке» - шкала отображается так, что значения галочки шкалы увеличиваются в по часовой стрелке.
- «против часовой стрелки» - шкала отображается так, что значения отметок шкалы увеличиваются против часовой стрелки.

5-123

## Страница 184

5 Ссылка на недвижимость

**Размер** - размер полукруглого датчика  
[120,65] (по умолчанию) | [ширина высота]

Размер полукруглого датчика, указанный как [ширина, высота]. Единицами измерения являются пиксели. Значение свойства Size изменяется при изменении ориентации.

**Значение** - расположение иглы  
0 (по умолчанию) | числовой

Расположение иглы полукруглого датчика, указывается в виде любого числового значения. Если значение меньше минимального значения свойства Limits, тогда стрелка указывает на местоположение непосредственно перед началом шкалы. Если значение больше максимального

Ограничивает значение свойства, затем стрелка указывает на местоположение сразу после окончания масштаб.

Изменение значения свойства Limits не влияет на настройку свойства Value.

Пример: 60

**Видимый** - видимость полукруглого датчика  
true (по умолчанию) | ложный

Видимость полукруглого датчика, указанная как истинная или ложная. Видимая собственность определяет, отображается ли полукруглый датчик на экране. Если видимый свойство полукруглого датчика установлено в значение false, весь полукруглый датчик скрыт, но вы все равно можете указать и получить доступ к его свойствам.

5-124

## Страница 185

## Свойства слайдера

Управление внешним видом и поведением слайдера

Слайдеры - это компоненты приложения, которые позволяют пользователям выбирать значение вдоль континуума. Обычно ползунки генерируют действие, когда пользователь приложения меняет значение. свойства контролировать внешний вид и поведение конкретного экземпляра слайдера. Модифицировать аспекты слайдера, изменение значений свойств. Используйте точечную запись для обозначения конкретного объекта и собственность:

```
слайдер = uislider;  
val = slider.Value;  
slider.Value = 20;
```

**Включено** - рабочее состояние ползунка

true (по умолчанию) | ложный

Рабочее состояние ползунка, указывается как true или false.

Если для этого свойства установлено значение true, пользователь приложения может изменить значение ползунка.

Если для этого свойства установлено значение false, ползунок отображается тусклым, указывая на то, что приложение Пользователь не может изменить значение ползунка, и ползунок не будет вызывать обратный вызов.

**FontAngle - Наклон** персонажа от основных меток слайдера

'нормальный' (по умолчанию) | «Наклонным»

Наклон символов меток основных тиканов ползунка, обозначенных как «обычный» или «курсив». настройка это свойство курсивом выбирает наклонную версию шрифта, если она доступна в приложении пользовательская система.

**FontName** - шрифт, в котором отображаются основные метки слайдера

'Helvetica' (по умолчанию) | строка

Шрифт, в котором отображается значение свойства MajorTickLabels ползунка, указанное в виде строки. Если указанный шрифт недоступен для пользователя приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

5-125

**FontSize** - размер шрифта для основных меток слайдера  
12 (по умолчанию) | положительное число

Размер шрифта меток основных галочек слайдера, указанный как положительное число. Единицы измерения это точки.

Пример: 14

**FontWeight** - толщина текстовых символов

'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**Limits** - минимальные и максимальные значения ползунка



[0,100] (по умолчанию) | массив  
Минимальное и максимальное значения ползунка, заданные в виде 2-элементного числового массива. Первый значение должно быть ниже второго значения.

**Расположение** - расположение ползунка, за исключением отметок и меток, относительно родительского [100,100] (по умолчанию) | [X, y]

Расположение ползунка, исключая отметки и метки, относительно родителя, указано как [x, y], где x и y определяют координаты из нижнего левого угла родительского контейнера в нижнем левом углу слайдера. Единицами измерения являются пиксели.

**MajorTicks - Значения** основных отметок  
[0, 20, 40, 60, 80, 100] (по умолчанию) | числовой вектор

Значения основных отметок, заданные в виде числового вектора. Любые значения массива, которые выходят за пределы Пределы не отображаются на слайдере. MATLAB удаляет дублирующиеся значения массива.

Установка значения свойства MajorTicks устанавливает значение свойства MajorTicksMode равным 'руководство по эксплуатации'.

**MajorTickLabels** - метки на каждой основной отметке  
{'0', '20', '40', '60', '80', '100'} (по умолчанию) | массив ячеек строк

5-126

Метки на каждой основной метке, указанной в виде массива строк. MATLAB использует каждый элемент в качестве метки для соответствующей основной отметки.

Установка MajorTickLabels изменяет значение MajorTickLabelsMode на «ручной».

Чтобы создать пустую метку для основного тика, укажите пустую строку для этого тика соответствующий элемент MajorTickLabels.

Если вы указали больше элементов MajorTickLabels, чем элементов MajorTicks, MATLAB игнорирует лишние метки.

Если вы указали меньше элементов MajorTickLabels, чем элементов MajorTicks, MATLAB оставляет лишние метки без надписей.

Пример: {'100', '80', '60', '40', '20', '0'}

Пример: {'Min', 'Max'}

**MajorTickLabelsMode** - режим создания меток **основных тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB создает массив строк для заполнения MajorTickLabels.  
MATLAB преобразует каждый элемент числового основного тика в строку, используя num2str.
- 'manual' - вы указываете массив строк MajorTickLabels.

**MajorTicksMode** - режим создания **основных тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Режим создания основного тика, указанный как один из следующих:

- 'auto' - MATLAB заполняет MajorTicks, создавая несколько основных отметок в диапазон, указанный свойством Limits (включительно).
- «ручной» - вы указываете массив значений MajorTicks.

**MinorTicks** - Незначительные значения отметки  
[0: 5: 100] (по умолчанию) | числовой вектор

Незначительные значения отметки, заданные в виде числового вектора. Любые элементы MinorTicks, которые падают внешние пределы не отображаются на датчике. Если незначительный тик падает на то же значение, что и основной тик, отображается только основной тик. MATLAB удаляет дубликаты MinorTicks

значения элемента. Установка значения свойства MinorTicks устанавливает MinorTicksMode  
Значение свойства «вручную».

**MinorTicksMode** - режим создания **мелких тиков**  
'auto' (по умолчанию) | 'руководство по эксплуатации'

Незначительный режим создания тиков, указанный как «автоматический» или «ручной».

Когда MinorTicksMode установлен в 'auto', MATLAB определяет размещение второстепенных  
тики, находя самый большой интервал между двумя последовательными основными тиками и делением  
этот интервал на 4. Например, если основные тики распределены равномерно, то ползунок  
будет иметь 3 небольших тика между каждым основным тиком.

**Ориентация** - Направление, в котором отображается слайдер  
'горизонтальный' (по умолчанию) | «Вертикальный»

Направление, в котором отображается слайдер, указано как «горизонтальное» или «вертикальное».

**OuterLocation** - расположение ползунка, включая метки и метки состояния, относительно родительского  
[93, 72] (по умолчанию) | [X, y]

Расположение ползунка, включая метки и метки состояний относительно родителя, возвращается как

[x, y], где x и y задают координаты из левого нижнего угла родителя  
Контейнер в нижнем левом углу слайдера. Единицами измерения являются пиксели.

**OuterSize** - размер **ползунка** , включая отметки и метки  
[136,40] (по умолчанию) | [ширина высота]

Размер ползунка, включая отметки и метки, возвращается как [ширина, высота].  
Единицами измерения являются пиксели.

**Родитель** - родительский контейнер слайдера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер ползунка, указанный в качестве дескриптора окна приложения, панели или группы кнопок  
объект.

**Размер** - размер слайдера, за исключением отметок и меток  
[120,6] (по умолчанию) | [ширина высота].

Размер ползунка, исключая отметки и метки состояния, определяется как [ширина, высота].  
Единицами измерения являются пиксели.

**Значение** - значение слайдера  
0 (по умолчанию) | числовой

Значение слайдера, указанное в числовом формате. Числовое значение должно быть в указанном диапазоне  
за пределы.

**ValueChangedFcn** - функция обратного вызова, которая выполняется один раз, когда пользователь приложения меняет ползунок  
стоимость  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова, которая выполняется один раз, когда пользователь приложения изменяет значение ползунка, заданное как  
одно из следующего:

- Функциональная ручка

- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пример: @function

Пример: {@function, x}

**ValueChangingFcn** - функция обратного вызова для выполнения, когда пользователь приложения изменяет значение ползунка  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова, выполняемая пользователем приложения, изменяет значение ползунка, указанное как единое целое, из следующих:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Когда пользователь приложения перетаскивает ползунок, MATLAB генерирует данные события и сохраняет их в Значение свойства события, к которому вы можете обратиться.

Например, чтобы отобразить значение события в командном окне MATLAB, укажите код в обратном вызове ValueChangingFcn:

```
дисп (event.Value)
```

Обратный вызов выполняется следующим образом:

- Если пользователь приложения щелкает, чтобы переместить ползунок к новому значению, обратный вызов выполняется один раз.

5-129

## Страница 190

5 Ссылка на недвижимость

Например, если ползунок на 1.0, а пользователь щелкает 1.1, обратный вызов выполняется, и значение события обновляется один раз до 1.1.

- Если пользователь приложения щелкает и перетаскивает ползунок к новому значению, выполняется обратный вызов несколько раз.

Например, если значение ползунка равно 1,0, а пользователь щелкает, удерживает и перетаскивает значение 1.1, выполняется обратный вызов и значение события обновляется до 1.1. Если пользователь продолжает перетаскивание до 1.2, обратный вызов выполняется снова, и значение события обновляется до 1.2. Когда пользователь отпускает мышь, обратный вызов не выполняется.

- Если свойство Slider Value изменяется программно, то обратный вызов не выполняется.

Количество отправляемых событий варьируется в зависимости от таких факторов, как скорость мыши, операционная система и разрешение экрана.

Пример: @function

Пример: {@function, x}

**Видимый** - видимость ползунка  
true (по умолчанию) | ложный

Видимость ползунка, указанная как true или false. Свойство Visible определяет, является ли слайдер отображается на экране. Если свойство Visible слайдера установлено в false, весь слайдер скрыт, но вы все равно можете указать и получить доступ к его свойствам.

## Свойства кнопки состояния

Управление состоянием кнопки появления и поведения

Кнопки состояния - это компоненты приложения, которые указывают логическое состояние. Как правило, кнопки состояния генерировать действие при изменении состояния. Свойства контролируют внешний вид и поведение определенного экземпляра кнопки состояния. Чтобы изменить аспекты кнопки состояния, изменить значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
кнопка = uibutton («состояние»);
selected = button.Value;
button.Value = true;
```

**Включено** - кнопка состояния рабочего состояния  
true (по умолчанию) | ложный

Рабочее состояние кнопки состояния, заданное как true или false.

Если для этого свойства задано значение true, кнопка состояния выглядит нормально и приложение пользователь может выбрать его.

Если вы установите для этого свойства значение false, внешний вид кнопки состояния будет затемнен, указывает на то, что пользователь приложения не может выбрать его.

**FontAngle** - символьный наклон текста кнопки состояния  
'нормальный' (по умолчанию) | «Наклонным»

Символьный уклон текста кнопки состояния, указанный как «обычный» или «курсив». Установка этого свойство курсивом выбирает наклонную версию шрифта, если она доступна пользователю приложения система.

**FontName** - шрифт для отображения текста кнопки состояния  
'Helvetica' (по умолчанию) | строка

Шрифт, в котором отображается кнопка состояния Значение свойства текста, заданное в виде строки. Если указанный шрифт недоступен для пользователя приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.  
Пример: «Ариал»

**FontSize** - размер шрифта текста кнопки состояния  
12 (по умолчанию) | положительное число

Размер шрифта текста кнопки состояния, указанный как положительное число. Единицы измерения являются точками.  
Пример: 14

**FontWeight** - толщина текстовых символов

'нормальный' (по умолчанию) | 'жирный'  
Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**HorizontalAlignment** - Горизонтальное выравнивание текста и значка внутри кнопки состояния  
'центр' (по умолчанию) | «левый» | 'правильно'

Горизонтальное выравнивание текста и значка внутри кнопки состояния, указанной как «центр»,  
'влево или вправо'.

На следующем рисунке показаны три варианта горизонтального выравнивания: по центру, слева и справа, когда для свойств VerticalAlignment и IconAlignment установлены значения по умолчанию ('center' и 'left' соответственно).

**Icon** - имя файла иконки для отображения на кнопке  
" (по умолчанию) | строка

Имя файла иконки для отображения на кнопке состояния, указанной в виде строки.

Строка должна быть именем и расширением файла изображения в пути MATLAB, или полный путь к файлу изображения.

5-132

Тип файла изображения должен быть JPEG, GIF или PNG.

MATLAB обрезает изображение, если не умещается в пределах, указанных кнопкой состояния  
Размер собственности.

Пример: 'icon.png'

Пример: 'C: \ Documents \ icon.png'

**IconAlignment** - расположение значка кнопки относительно текста  
'left' (по умолчанию) | «центр» | «верх» | «дно» | 'правильно'

Расположение значка кнопки состояния относительно кнопки состояния Текст (если есть), указанный как один из следующий:

Стоимость	Результат
'оставил'	
'центр'	
'Вверх'	
'низ'	
'правильно'	

Примечание. Предыдущие изображения отражают расположение значков при горизонтальном выравнивании и Каждому VerticalAlignment присваивается значение по умолчанию, «center».

Если вы укажете пустую строку (") для свойства Text, свойство IconAlignment

не имеет никакого эффекта

**Расположение** - расположение кнопки состояния относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Расположение кнопки состояния относительно родительского контейнера, указывается как [x, y], где x и y укажите координаты от нижнего левого угла родительского контейнера до нижнего левый угол кнопки состояния. Единицами измерения являются пиксели.

5-133

**OuterLocation** - расположение кнопки состояния относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично расположению для кнопок состояния.

**OuterSize** - размер кнопки состояния  
[100,20] (по умолчанию) | [ширина высота]

Идентичен размеру для кнопок состояния.

**Родитель** - Родительский контейнер кнопки состояния  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер кнопки состояния, указанный как окно приложения, панель или группа кнопок объект.

**Размер** - размер кнопки состояния  
[100,20] (по умолчанию) | [ширина высота]

Размер кнопки состояния, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Текст** - текст кнопки состояния  
«Кнопка состояния» (по умолчанию) | строка

Текст кнопки состояния, указанный в виде строки.

Пример: «Устойчивое состояние»

**Значение** - нажата ли кнопка состояния  
true (по умолчанию) | ложный

Нажата ли кнопка состояния, указано как true или false.

**ValueChangedFcn** - функция обратного вызова, которая выполняется, когда пользователь приложения нажимает или отпускает кнопка состояния  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова, которая выполняется, когда пользователь приложения нажимает или отпускает кнопку состояния, указано как одно из следующих:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы

5-134

• Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пример: @function

Пример: {@function, x}

**VerticalAlignment** - Вертикальное выравнивание текста внутри кнопки состояния  
'центр' (по умолчанию) | «верх» | 'низ'

Вертикальное выравнивание текста внутри кнопки состояния. указано как «центр», «верх» или 'низ'.

На следующем рисунке показаны три варианта вертикального выравнивания, когда Горизонтальное выравнивание является «центром». На изображении сверху вниз вертикальная Значения выравнивания: «верх», «центр», «низ».

Примечание. Значения вертикального выравнивания не влияют, когда кнопка высота по умолчанию.

**Видимый** - видимость кнопки состояния  
true (по умолчанию) | ложный

Видимость кнопки состояния, указанная как true или false. Видимое свойство определяет отображается ли кнопка состояния на экране. Если видимая собственность государства Кнопка имеет значение false, кнопка состояния полностью скрыта, но вы все равно можете указать и получить доступ к его свойствам.

5-135

## Переключить свойства

Внешний вид и поведение переключателя

Переключатели - это компоненты приложения, которые указывают логическое состояние. Как правило, переключатели генерируют действие, когда состояние меняется. Свойства контролируют внешний вид и поведение конкретный экземпляр переключателя. Чтобы изменить аспекты флажка, измените значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
ползунковый переключатель = uiswitch;
val = sliderswitch.Value;
sliderswitch.Value = true;
```

**Включено** - рабочее состояние коммутатора  
true (по умолчанию) | ложный

Рабочее состояние переключателя, указывается как true или false.

Если для этого свойства установлено значение true, пользователь приложения может сдвинуть переключатель.

Если для этого свойства установлено значение false, переключатель будет недоступен, что означает, что приложение пользователь не может сдвинуть переключатель, и он не будет вызывать обратный вызов.

**Location** - расположение переключателя, исключая метки состояния относительно родительского [121,100] (по умолчанию) | [x, y]

Расположение переключателя относительно родителя, указанного как [x, y], где x и y определяют координаты от нижнего левого угла родительского контейнера до нижнего левого угла Переключатель. Единицами измерения являются пиксели.

**Ориентация** - Направление, в котором отображается переключатель  
'горизонтальный' (по умолчанию) | «Вертикальный»

Направление, в котором отображается переключатель, указывается как «горизонтальный» или «вертикальный».

**OuterLocation** - расположение переключателя, включая метки состояния, относительно родительского [100,100] (по умолчанию) | [X, y]

5-136

Расположение переключателя, включая метки состояния, относительно родителя, возвращается как [x, y], где x и y задают координаты от левого нижнего угла родительского контейнера до нижний левый угол переключателя, включая метки состояния. Единицами измерения являются пиксели.

**OuterSize** - размер переключателя, включая метки состояния  
[88,20] (по умолчанию) | [ширина высота]

Размер переключателя, включая метки состояния, возвращается как [ширина, высота]. Единицы измерения в пикселях.

**Родитель** - Родительский контейнер коммутатора  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер переключателя, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - размер переключателя, без учета государственных меток  
[45,20] (по умолчанию) | [ширина высота]

Размер переключателя, исключая метки состояния, определяется как [ширина, высота]. Единицы измерения в пикселях.

**Текст** - переключение меток состояния  
{'Off', 'On'} (по умолчанию) | 1 на 2 строки массива ячеек

Переключите метки состояния, указанные как массив строк размером 1 на 2.

Когда свойство переключателя Orientation установлено на «вертикальный», первый элемент массив появляется в нижней части коммутатора. Когда свойство Orientation установлено в «горизонтальный», первый элемент массива появляется слева от переключателя.

**Значение** - состояние переключателя  
false (по умолчанию) | правда

Состояние переключателя, указанное как одно из следующих значений:

- ложный

Переключатель направлен вниз, когда свойство Orientation установлено на «вертикальное» и указывает влево, когда свойство Orientation установлено на «горизонтальный».

- правда

5-137



Переключатель направлен вверх, когда для свойства Orientation установлено значение 'vertical', и указывает право, когда свойство Orientation установлено в значение «горизонтальный».

**ValueChangedFcn** - код для выполнения, когда пользователь приложения меняет состояние переключателя  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Код для выполнения, когда пользователь приложения меняет состояние переключателя, указанное в качестве одного из следующих значений:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пользователь приложения может изменить состояние переключателя, нажав и отпустив кнопку мыши в любом месте на коммутаторе (включая метки состояния) или нажав на коммутатор, перетаскивая и отпуская кнопку мыши, находясь все еще на переключателе.

Пример: @function

Пример: {@function, x}

**Видимый** - Переключение видимости  
true (по умолчанию) | ложный

Переключить видимость, указанную как true или false. Свойство Visible определяет, является ли переключатель отображается на экране. Если свойство Visible переключателя установлено в false, весь переключатель скрыт, но вы все равно можете указать и получить доступ к его свойствам.

## Свойства текстовой области

Контроль внешнего вида и поведения текстовой области

Текстовые области - это компоненты приложения, которые позволяют пользователям приложения указывать несколько строк текста. Как правило, текстовые области генерируют действие после того, как пользователь приложения вводит текст. Управление свойствами внешний вид и поведение конкретного экземпляра текстовой области. Чтобы изменить аспекты текстовая область, измените значения свойств. Используйте точечную запись для обозначения конкретного объекта и свойство:

```
textarea = uitablearea;  
text = textarea.Value;  
textarea.Value = {'Джозеф Уэлфорд'; 'Мэри Рейли'; 'Роберта Сильберлинг'};
```

## Стиль шрифта

**Редактируемый - редактируется** ли текстовая область  
true (по умолчанию) | ложный

Является ли текстовая область редактируемой, указана как true или false.

В этой таблице приведены визуальные и функциональные результаты различных комбинаций значения свойств для редактируемых и включенных свойств:

Включено	Редактирование	Визуальные результаты	Функциональные результаты
'правда'	'правда'		Пользователь приложения может обновить текст поле. Обратный вызов, связанный с текстовым полем будет выполнен при изменении текста.
'ложный'	'ложный'		Пользователь приложения не может обновить текстовое поле.

Страница 200

5 Ссылка на недвижимость

Включено	Редактирование	Визуальные результаты	Функциональные результаты
'правда'	'ложный'		Пользователь приложения не может обновить текстовое поле.
'ложный'	'правда'		Пользователь приложения не может обновить текстовое поле.

**Включено** - рабочее состояние текстовой области  
true (по умолчанию) | ложный

Рабочее состояние текстовой области, заданное как true или false.

Если для этого свойства установлено значение true, текст в текстовой области выглядит нормально.

Если для этого свойства установлено значение false, появляется текст в текстовой области. затемнено, указывая, что текстовая область не может быть обновлена и не будет вызывать обратный вызов.

В этой таблице приведены визуальные и функциональные результаты различных комбинаций значения свойств для редактируемых и включенных свойств:

Включено	Редактирование	Визуальные результаты	Функциональные результаты
'правда'	'правда'		Пользователь приложения может обновить текст поле. Обратный вызов, связанный с текстовым полем будет выполнен при изменении текста.
'ложный'	'ложный'		Пользователь приложения не может обновить текстовое поле.
'правда'	'ложный'		Пользователь приложения не может обновить текстовое поле.

Включено	Редактирование	Визуальные результаты	Функциональные результаты
'ложный'	'правда'		Пользователь приложения не может обновить текстовое поле.

**FontAngle** - символьный уклон текстовой области текста  
'нормальный' (по умолчанию) | «Наклонным»

Символьный уклон текста текстовой области, указанный как «обычный» или «курсив». Установка этого свойства курсивом выбирает наклонную версию шрифта, если она доступна пользователю приложения система.

**FontName** - шрифт для отображения текста в текстовой области  
'Helvetica' (по умолчанию) | строка

Шрифт для отображения в текстовой области значения свойства Value, указанного в виде строки. Если указанный шрифт недоступен пользователю приложения, MATLAB использует Helvetica. Если Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

**FontSize** - размер шрифта текста текстовой области  
12 (по умолчанию) | положительное число

Размер шрифта текста текстовой области, указанный как положительное число. Единицы являются точками.

Пример: 14

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**HorizontalAlignment** - Горизонтальное выравнивание текста в текстовой области  
'left' (по умолчанию) | «правильно» | 'центр'

Выравнивание текста в текстовой области, указанной как «влево», «вправо» или «по центру».

Выравнивание влияет на отображение, так как пользователь приложения редактирует текстовую область и то, как MATLAB отображает текст в приложении.

**Местоположение** - расположение текстовой области относительно родителя  
[100,100] (по умолчанию) | [X, y]

Расположение текстовой области относительно родителя, указывается как [x, y], где x и y определяют координаты от нижнего левого угла родительского контейнера до нижнего левого угла текстовая область. Единицами измерения являются пиксели.

**OuterLocation** - расположение текстовой области относительно родительского контейнера.  
[100,100] (по умолчанию) | [X, y]

Идентично местоположению для текстовых областей.

**OuterSize** - размер текстовой области, включая значение **текста**  
[150,50] (по умолчанию) | [ширина высота]

Идентичен по размеру для текстовых областей.

**Родитель** - Родительский контейнер текстовой области  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер текстовой области, указанный как окно приложения, панель или объект группы кнопок.

**Размер** - Размер текстовой области  
[150,50] (по умолчанию) | [ширина высота]

Размер области текста, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Значение** - текст в текстовой области  
{""} (по умолчанию) | массив ячеек строк

Текст в текстовой области, указанный как массив ячеек строк. MATLAB может правильно отображать форматированный текст, такой как этот:

```
cellArrayText {1} = sprintf( '%s \n %s', 'Строка 1', 'Строка 2' )
cellArrayText {2} = sprintf( '%s \n %s', 'Строка 3', 'Строка 4' )
textarea = uitablearea ( 'Value', cellArrayText);
```

5-142

Если строка текста в массиве ячеек не помещается в ширину области текста, MATLAB переносит текст

Если в текстовой области слишком много строк для отображения, MATLAB добавляет полосу прокрутки.

Пример: {'Джозеф Уэлфорд'; «Мэри Рейли»; 'Роберта Силберлихт'}

**ValueChangedFcn** - обратный вызов, чтобы выполнить после того, как пользователь приложения фиксирует ввод текста  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Функция обратного вызова, выполняемая после ввода пользователем приложения текста, указанного в качестве дескриптора функции, массив ячеек, содержащий дескриптор функции и дополнительные аргументы, или строку.

Обратный вызов выполняется после того, как пользователь приложения набирает текст в текстовой области, а затем нажимает

**Введите** или щелкните за пределами текстовой области.

Пример: @function

Пример: {@function, x}

**Видимый** - видимость текстовой области  
true (по умолчанию) | ложный

Видимость области текста, указанная как true или false. Видимое свойство определяет отображается ли текстовая область на экране. Если свойство Visible текстовой области имеет значение false, вся текстовая область скрыта, но вы все равно можете указать и получить доступ к ее свойствам.

---

Страница 2045 [Ссылка на недвижимость](#)

## Свойства кнопки переключения

Управление переключателем внешнего вида

Кнопки переключения - это компоненты приложения, которые отображаются в группе кнопок как набор эксклюзивные варианты. Как правило, группа кнопок, к которой принадлежит кнопка переключения генерирует действие при изменении выбора. Свойства контролируют внешний вид и поведение конкретного экземпляра кнопки переключения. Чтобы изменить аспекты кнопки переключения, изменить значения свойств. Используйте точечную нотацию для ссылки на конкретный объект и свойство:

```
togbutton = uitogglebutton;  
txt = togbutton.Text;  
togbutton.Text = 'Гистограмма';
```

Примечание. Чтобы указать, как ваше приложение должно реагировать на изменение пользователем кнопки переключения. выделение, код обратного вызова `ValueChangedFcn` для группы кнопок, к которой переключается кнопки являются родителями.

**Включено** - рабочее состояние кнопки переключения

true (по умолчанию) | ложный

Рабочее состояние кнопки переключения, заданное как true или false.

Если вы установите для этого свойства значение true, появление кнопки переключения означает, что Пользователь приложения может выбрать его.

Если вы установите для этого свойства значение false, внешний вид кнопки переключения будет затемнен, указывает на то, что пользователь приложения не может выбрать его.

**FontAngle** - символьный наклон текста переключателя

'нормальный' (по умолчанию) | «Наклонным»

Наклон символов текста кнопки переключения, заданный как «обычный» или «курсив». настройка это свойство курсивом выбирает наклонную версию шрифта, если она доступна в приложении пользовательская система.

**FontName** - шрифт, в котором отображается текст кнопки переключения

'Helvetica' (по умолчанию) | строка

5-144

---

Страница 205

Свойства кнопки переключения

Шрифт, в котором отображается кнопка-переключатель Значение свойства текста, заданное в виде строки.

Если указанный шрифт недоступен для пользователя приложения, MATLAB использует Helvetica. Если

Helvetica недоступна, MATLAB использует доступный шрифт без засечек.

Пример: «Ариал»

**FontSize** - размер шрифта текста кнопки переключения

12 (по умолчанию) | положительное число

Размер шрифта текста кнопки-переключателя, указанный как положительное число. Единицы измерения являются точками.

Пример: 14

**FontWeight** - толщина текстовых символов  
'нормальный' (по умолчанию) | 'жирный'

Толщина текстовых символов, указанная в качестве одного из следующих значений:

- 'normal' - вес по умолчанию, определенный конкретным шрифтом
- «жирный» - более толстые контуры символов, чем «нормальные»

Не все шрифты имеют жирный шрифт. Следовательно, указание веса жирного шрифта может привести к в нормальном весе шрифта.

**HorizontalAlignment** - Горизонтальное выравнивание значка и текста внутри кнопки переключения  
'центр' (по умолчанию) | «левый» | 'правильно'

Горизонтальное выравнивание текста и значка внутри кнопки переключения, указанной как «центр», 'влево или вправо'.

На следующем рисунке показаны три варианта горизонтального выравнивания: по центру, слева и справа, когда для свойств VerticalAlignment и IconAlignment установлены значения по умолчанию ('center' и 'left' соответственно).

**Icon** - имя файла иконки для отображения на кнопке  
" (по умолчанию) | строка

5-145

Имя файла иконки для отображения на кнопке переключения, указанной в виде строки.

Строка должна быть именем и расширением файла изображения в пути MATLAB, или полный путь к файлу изображения.

Тип файла изображения должен быть JPEG, GIF или PNG.

MATLAB обрезает изображение, если не умещается в пределах, указанных кнопкой переключения  
Значение свойства размера.

Пример: 'icon.png'

Пример: 'C:\Documents \ icon.png'

**IconAlignment** - расположение значка кнопки относительно текста  
'left' (по умолчанию) | «центр» | «верх» | «дно» | 'правильно'

Расположение значка переключателя относительно переключателя «Текст» (если есть), указанного как один из последующий:

Стоимость	Результат
'оставил'	
'центр'	
'Вверх'	
'низ'	
'правильно'	

Примечание. Предыдущие изображения отражают расположение значков при горизонтальном выравнивании и Каждому VerticalAlignment присваивается значение по умолчанию, «center».

Если вы укажете пустую строку (") для свойства Text, то значок будет выровнен, как если бы это текст, в соответствии со значениями, установленными для HorizontalAlignment и VerticalAlignment. Значение свойства IconAlignment в этом случае не влияет.

5-146

---

## Страница 207

Свойства кнопки переключения

**Расположение** - расположение кнопки переключения относительно группы кнопок  
[10,10] (по умолчанию) | [X, y]

Расположение кнопки переключения относительно контейнера группы родительских кнопок, заданное как

[x, y], где x и y определяют координаты из нижнего левого угла кнопки группа контейнеров в нижнем левом углу кнопки переключения. Единицы измерения являются пикселями.

**OuterLocation** - расположение кнопки переключения относительно родительского контейнера.  
[10,10] (по умолчанию) | [X]

Идентично расположению для кнопок переключения.

**OuterSize** - размер кнопки переключения  
[100,20] (по умолчанию) | [ширина высота]

Идентичен по размеру для кнопок переключения.

**Parent** - контейнер группы родительских кнопок для кнопки-переключателя  
объект группы кнопок

Контейнер группы родительских кнопок для кнопки-переключателя, указанный как объект группы кнопок. Видеть  
Группа кнопок для деталей.

**Размер** - размер кнопки переключения  
[100,20] (по умолчанию) | [ширина высота]

Размер кнопки переключения, указанный как [ширина, высота]. Единицами измерения являются пиксели.

**Текст** - текст кнопки переключения  
«Кнопка переключения» (по умолчанию) | строка

Текст кнопки переключения, указанный в виде строки.

Пример: «Устойчивое состояние»

**Значение** - состояние выбора кнопки переключения  
правда | ложный

Состояние выбора кнопки переключения, заданное как true или false.

В пределах данной группы кнопок переключения может быть выбрана (нажата) только одна кнопка время.

5-147

---

## Страница 208

5 Ссылка на недвижимость

Когда свойство Value для кнопки переключения меняется на true, свойство Value для ранее выбранная кнопка переключения, принадлежащая той же группе кнопок, становится ложной. Кроме того, значение свойства SelectedObject группы кнопок обновляется в соответствии с выбранная кнопка переключения.

Если вы программно измените свойство Value для кнопки-переключателя на false,

MATLAB устанавливает для первой кнопки переключения в группе кнопок значение true. Если первый переключатель кнопка, для которой вы устанавливаете свойство Value для переключателя на false, затем MATLAB устанавливает вторую кнопку переключения в группе кнопок на true.

**VerticalAlignment** - Вертикальное выравнивание текста внутри кнопки переключения  
'центр' (по умолчанию) | «верх» | 'низ'

Вертикальное выравнивание текста внутри кнопки переключения. указано как «центр», «верх» или 'низ'.

Примечание. Значения вертикального выравнивания, по-видимому, не действуют, если кнопка переключения высота по умолчанию.

**Видимый** - Видимость кнопки переключения  
true (по умолчанию) | ложный

Переключатель видимости кнопки, заданный как true или false. Видимое свойство определяет отображается ли кнопка переключения на экране.

## Свойства тумблера

Тумблер внешнего вида и поведения

Тумблеры - это компоненты приложения, которые указывают на логическое состояние. Как правило, переключатель переключатели генерируют действие при изменении состояния. Свойства контролируют внешний вид и поведение конкретного экземпляра тумблера. Чтобы изменить аспекты переключения переключить, изменить значения свойств. Используйте точечную запись для обозначения конкретного объекта и свойство:

```
togswitch = uiswitch('toggle');  
val = togswitch.Value;  
togswitch.Value = true;
```

**Включено** - рабочее состояние тумблера  
true (по умолчанию) | ложный

Рабочее состояние тумблера, заданное как true или false.

Если для этого свойства установлено значение true, конечный пользователь может переключать переключатель.

Если вы установите для этого свойства значение false, тумблер будет затемнен, указывая на то, что Пользователь приложения не может переключить переключатель, и он не будет вызывать обратный вызов.

**Расположение** - расположение тумблера, исключая метки состояния, относительно родителя



[121 100] (по умолчанию) | [X, y]  
Расположение тумблера, исключая метки состояния, относительно родителя, указанного как [x, y], где x и y определяют координаты из нижнего левого угла родительского контейнера в нижний левый угол тумблера. Единицами измерения являются пиксели.

**Ориентация** - Направление, в котором отображается тумблер  
'вертикальный' (по умолчанию) | «Горизонтальные»  
Направление, в котором отображается тумблер, указано как «горизонтальное» или «Вертикальный».

5-149

**OuterLocation** - Расположение тумблера, включая метки состояния, относительно родителя.  
[100,100] (по умолчанию) | [X, y]  
Расположение тумблера, включая метки состояния, относительно родителя, возвращается как [x, y], где x и y задают координаты из левого нижнего угла родителя  
Контейнер до левого нижнего угла тумблера, включая метки состояния. Единицы измерения в пикселях.

**OuterSize** - размер тумблера, включая метки состояния  
[20,87] (по умолчанию) | [ширина высота]  
Размер тумблера, включая метки состояния, возвращается как [ширина, высота]. Единицы измерения являются пикселями.

**Parent** - родительский контейнер тумблера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок  
Родительский контейнер тумблера, указанный как окно приложения, панель или группа кнопок объект.

**Размер** - размер тумблера, без меток состояния  
[20,45] (по умолчанию) | [ширина высота]  
Размер тумблера, исключая метки состояния, определяется как [ширина, высота].

**Текст** - метки состояния переключателя  
{'Off', 'On'} (по умолчанию) | Строки массива ячеек 1 на 2  
Метки состояния тумблера, указанные как массив строк размером 1 на 2.  
Когда свойство ориентации переключателя установлено на «вертикальный», первый элемент из массива появляется в нижней части тумблера. Когда свойство Ориентация установлен на «горизонтальный», первый элемент массива появляется слева от переключателя переключатель.

**Значение** - состояние тумблера  
false (по умолчанию) | правда  
Состояние тумблера, заданное одним из следующих значений:

- ложный

5-150

Тумблер указывает вниз, когда свойство Orientation установлено на «вертикальное»

и указывает влево, когда для свойства Orientation установлено значение «горизонтальный».

- правда

Тумблер указывает вверх, когда свойство Orientation установлено на «вертикальное» и указывает вправо, когда свойство Orientation установлено на «горизонтальный».

**ValueChangedFcn** - код, который выполняется, когда пользователь приложения меняет состояние переключателя  
[] (по умолчанию) | ручка функции | массив ячеек | строка

Код, выполняемый при изменении пользователем приложения состояния тумблера, указанного как один из следующий:

- Функциональная ручка
- Массив ячеек, содержащий дескриптор функции и дополнительные аргументы
- Строка, которая является допустимым выражением MATLAB, которое оценивается в базовой рабочей области.

Пользователь приложения может изменить состояние переключателя, нажав и отпустив мышью в любом месте на тумблере (включая метки состояния) или нажав на тумблер, перетаскивание и отпускание кнопки мыши, когда он еще находится на тумблере.

Пример: @function

Пример: {@function, x}

**Видимый** - видимость тумблера

true (по умолчанию) | ложный

Переключатель видимости, заданный как true или false. Видимое свойство определяет отображается ли тумблер на экране. Если свойство Visible переключателя переключатель имеет значение false, весь тумблер скрыт, но вы все равно можете указать и получить доступ к его свойствам.

5-152

---

Страница 213

6

## Справочник по функциям

[appdesigner](#)  
[appwindow](#)  
[осн](#)  
[UIButton](#)  
[uicheckbox](#)  
[uidropdown](#)  
[uieditfield](#)  
[uigauge](#)  
[uiknob](#)  
[UILabel](#)  
[uilamp](#)  
[uiliblistbox](#)  
[uiradiobutton](#)  
[uislider](#)  
[uiswitch](#)  
[uitextarea](#)  
[uitogglebutton](#)

---

Страница 214

# appdesigner

Создание или редактирование файла приложения в App Designer

## Синтаксис

appdesigner  
файл appdesigner

## Описание

appdesigner открывает среду разработки приложений, App Designer.

Файл appdesigner открывает указанный файл приложения в App Designer.

дизайнер приложений myappfile

## Входные аргументы

**file** - имя файла приложения  
строка

Имя файла приложения, указанное в виде строки. Если файл перегружен (то есть он появляется в несколько папок в пути поиска), затем укажите путь к файлу. Спецификация файла должен быть полный или относительный путь к файлу или имя файла в MATLAB дорожка. В противном случае MATLAB возвращает ошибку.

Пример: appdesigner App1

## Примеры

Откройте новый файл приложения

appdesigner

6-2

Новый файл с именем App1.mlapp откроется в конструкторе приложений. App1.mlapp не появляется в браузер MATLAB Current Folder, пока вы не сохраните файл.

Откройте существующий файл приложения

стоимость [проекта](#)

App Designer открывает и отображает существующий файл valueplots.mlapp, предполагая, что Файл находится в текущей папке или в пути поиска MATLAB. В противном случае MATLAB возвращает ошибка файла не найдена.

---

**Страница 216**

6 Справочник по функциям

## appwindow

Создать компонент окна приложения

### Синтаксис

```
appwindow  
window = appwindow (Имя, Значение)
```

### Описание

appwindow создает новое окно приложения, используя значения свойств по умолчанию.

window = appwindow (Name, Value) определяет свойства окна приложения, используя один или больше имени, значения пары аргументов.

### Входные аргументы

#### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список и описание свойств (аргументы пары имя-значение) см. В окне приложения. Свойства.

Пример: «Имя», «Результаты»

### Примеры

Окно создания приложения по умолчанию

```
appwindow;
```

---

**Страница 217**

appwindow

Доступ к свойствам окна приложения

Создайте окно приложения с указанным заголовком.

```
window = appwindow ( 'Name' , 'Plotted Results' );
```

**6-5**

---

**Страница 218****6** Справочник по функциям

Получите в окне приложения значение свойства Position.

```
position = window.Position
```

ANS =

```
680 678 560 420
```

Смотрите также

свойства  
Свойства окна приложения

6-6

---

Страница 219

оси

## ОСИ

Создание осей графического объекта

## Синтаксис

```
ax = axes ('Parent', приложение)  
ax = axes ('Parent', app, Name, Value)
```

## Описание

`ax = axes ('Parent', app)` создает объект оси в указанном окне приложения.

`ax = axes ('Parent', app, Name, Value)` определяет свойства осей, используя один или несколько Имя, Значение пары аргументов.

## Входные аргументы

**приложение** - окно приложения  
объект окна приложения

Окно приложения, указанное как объект окна приложения.

## Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты (' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список свойств см. В разделе Свойства осей.

Пример: «Цвет», «красный» определяет красный цвет для задних плоскостей осей.

6-7

---

## Страница 220

6 Справочник по функциям

## Примеры

Несколько осей в одном окне приложения

Создание и нанесение на несколько осей в одном окне приложения.

Укажите объект axes в команде plot, чтобы указать, по каким осям вы хотите каждый график быть отображенным.

```
window = appwindow;  
a = [1,4,7,9];  
b = [10,12,30,4];  
ax1 = axes ( 'Parent' , окно, 'Units' , 'пиксели' , ...  
            «Положение» , [67, 171, 175, 152]);  
ax2 = axes ( 'Parent' , окно, 'Units' , 'пиксели' , ...  
            «Должность» [330,167,179,156]);  
участок (ax1, a, b);  
участок (ax2, b, a);
```

6-8

---

## Страница 221

оси

Измените цвет задней плоскости ax2.

```
ax2.Color = 'red' ;
```



Определите шкалу значений вдоль оси x в оси 1.

```
xscale = ax1.XScale
```

ANS =

линейный

## Больше о

подсказки

- Функция оси (не оси) обеспечивает упрощенный доступ к часто используемым свойствам, которые контролируют масштабирование и внешний вид осей.

6-9

---

## Страница 222

6 Справочник по функциям

- Для получения информации о настройке свойств осей по умолчанию см. Раздел справки MATLAB, «Значения свойств по умолчанию».

## Смотрите также

функции

ось | кла | CLF | GCA | сетка | участок | название | посмотреть | xlabel | ylabel

свойства

Свойства осей

# UIButton

Создать кнопку или компонент кнопки состояния

## Синтаксис

```
кнопка = uibutton
кнопка = uibutton (стиль)
кнопка = uibutton (родитель)
кнопка = uibutton (родитель, стиль)
кнопка = uibutton (__, имя, значение)
```

## Описание

button = uibutton создает кнопку в новом окне приложения.

Кнопка = uibutton (стиль) создает кнопку указанного стиля.

button = uibutton (parent) указывает объект, в котором нужно создать кнопку.

button = uibutton (parent, style) создает кнопку указанного стиля в указанный родительский объект.

button = uibutton (\_\_, Name, Value) определяет свойства кнопки, используя один или больше имени, значения пары аргументов. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**стиль** - тип кнопки  
'толкать' (по умолчанию) | 'государство'

Тип кнопки, указанный как один из следующих:

- 'От себя'

При нажатии один раз, кнопка появляется, чтобы нажать и отпустить.

- 'государство'

При однократном нажатии кнопка остается в нажатом или отпущенном состоянии до нажатия снова.

**Родитель** - Родительский контейнер кнопки  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер кнопки, указанный как окно приложения, панель или объект группы кнопок.

Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' ). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список стилей и описание свойств (аргументы пары имя-значение) см. В разделе Свойства кнопки и свойства кнопки состояния.

Пример: «Текст», «Старт» указывает, что текст «Старт» отображается на кнопке.

Примеры

Создать кнопку

Создать кнопку, стиль по умолчанию для кнопки.

кнопка = uibutton;

Создать кнопку состояния

Создайте кнопку состояния, указав стиль как состояние.

кнопка = uibutton ( «состояние» );

Укажите родительский объект для кнопки

Укажите окно приложения в качестве родительского объекта для кнопки.

6-13

## Страница 226

6 Справочник по функциям

Создайте два окна приложения, app1 и app2. Укажите app2 в качестве родительского окна приложения для кнопка.

```
window1 = appwindow ( 'Name', 'One' );
window2 = appwindow ( 'Name', 'Two' );
кнопка = uibutton (окно 2);
```

Доступ к значениям свойства кнопки состояния

Создайте кнопку состояния и укажите значения свойств.

```
кнопка = uibutton ( 'состояние' , ...
    «Текст» , «Запись» , ...
    «Ценность» , правда , ...
    «Местоположение» , [50,100]);
```

Определите размер шрифта текста кнопки состояния:

```
fname = button.FontName
```

```
fname =
```

```
Helvetica
```

Изменить шрифт текста кнопки:

6-14

## Страница 227

UIButton

```
button.FontName = 'Arial Narrow' ;
```

Код кнопки обратного вызова ButtonPressedFcn для кнопки

Создайте окно приложения, содержащее кнопку и оси. Код обратного вызова для push кнопка, так что когда пользователь нажимает кнопку, MATLAB отображает указанные данные в оси

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция buttonPlot
% Создать окно приложения
window = appwindow;

% Создание осей
ax = axes ('Parent', окно, ...
           'Единицы измерения', 'пикселей', ...
           «Положение», [104, 123, 198, 201]);

% Создать кнопку
кнопка = uibutton (окно, «нажать», ...
                  «Местоположение», [384, 218], ...
                  'ButtonPhedFcn', @ (кнопка, событие) plotButtonPush (кнопка, топор));
конец

% Создать функцию для обратного вызова ButtonPressedFcn
функция plotButtonPush (кнопка, топор)
    x = пробел (0,2 * пи, 100);
    y = грех (x);
    участок (ax, x, y)
конец
```

Запустите buttonPlot, а затем нажмите кнопку.

Смотрите также

свойства  
Свойства кнопки | Свойства кнопки состояния

6-16

uicheckbox

# uicheckbox

Создать флажок компонента

## Синтаксис

checkbox = uicheckbox  
checkbox = uicheckbox (родительский)  
checkbox = uicheckbox (\_\_, имя, значение)

## Описание

checkbox = uicheckbox создает флажок в новом окне приложения.

checkbox = uicheckbox (parent) указывает объект, в котором нужно создать чек-коробка.

checkbox = uicheckbox (\_\_, Name, Value) определяет свойства флажка, используя один или более Имя, Аргументы пары Значение. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**parent** - флажок родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Установите флажок родительский контейнер, указанный как объект окна приложения, панели или группы кнопок.

## Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного кавычки (' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

6-17

Список и описание свойств (аргументы пары имя-значение) см. В разделе «Флажок». Свойства.

Пример: «Значение», true указывает, что флажок отображается с галочкой.

Примеры

Создать флажок

```
checkbox = uicontrol;
```

Укажите родительский объект для флажка

```
window = appwindow;  
checkbox = uicontrol (окно);
```

Флажок Доступа

Создайте флажок и укажите значения свойств.

6-18

uicontrol

```
checkbox = uicontrol ( 'Текст' , 'Показать значение' , ...  
                  «Ценность» , правда, ...  
                  «Местоположение» , [150,50]);
```

Снимите флажок:

```
checkbox.Value = false;
```

Определите размер шрифта текста флажка:

```
fsz = checkbox.FontSize
```

```
fsz =
```

```
12
```

Кодировать флажок ValueChangedFcn Callback

Создайте окно приложения, содержащее флажок и группу кнопок, содержащую три радио кнопки. Закодируйте обратный вызов для флажка так, чтобы, когда пользователь выбирает проверку

6-19

## Страница 232

6 Справочник по функциям

поле, третья кнопка в группе кнопок отключена. Когда пользователь снимает флажок, переключатель снова активирован.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция disableRadioButton
% Создать окно приложения:
window = appwindow ( 'Position' , [100, 100, 229, 276]);

% Создание группы кнопок и переключателей:
buttongroup = matlab.ui.control.ButtonGroup ( 'Parent' , окно, ...
    «Местоположение» , [56, 77]);
rb1 = uiradiobutton (buttongroup, 'Location' , [10,60]);
rb2 = uiradiobutton (buttongroup, 'Location' , [10,38]);
rb3 = uiradiobutton (buttongroup, 'Location' , [10,16]);

% Создать флажок:
uicheckbox (окно, 'Местоположение' , [55, 217], ...
    'ValueChangedFcn' , @ (флажок, событие) cBoxChanged (флажок, rb3));
конец

% Создайте функцию для обратного вызова ValueChangedFcn:
функция cBoxChanged (флажок, rb3)
val = checkbox.Value;
rb3.Enabled = ~ val;
конец
```

Запустите disableRadioButton, а затем снимите флажок.



uicheckbox

Смотрите также

свойства  
Флажок Свойства

# uidropdown

Создать выпадающий компонент

## Синтаксис

dropdown = uidropdown  
выпадающий = uidropdown (стиль)  
dropdown = uidropdown (родительский)  
dropdown = uidropdown (родитель, стиль)

dropdown = uidropdown ( \_\_\_, имя, значение)

### Описание

dropdown = uidropdown создает раскрывающийся список в новом окне приложения.

dropdown = uidropdown (style) создает раскрывающийся список указанного стиля.

dropdown = uidropdown (parent) указывает объект, в котором нужно создать каплю вниз.

dropdown = uidropdown (parent, style) создает раскрывающийся список указанного стиля в указанном родительском объекте.

dropdown = uidropdown ( \_\_\_, Name, Value) указывает свойства раскрывающегося списка используя один или несколько аргументов Имя, Пара значений. Используйте эту опцию с любым входом комбинации аргументов в предыдущих синтаксисах.

### Входные аргументы

**стиль** - Тип выпадающего  
«выпадающий» (по умолчанию) | «Редалируемые»

Тип раскрывающегося списка, указанный как один из следующих:

- 'падать'

6-22

uidropdown

Позволяет пользователю выбирать из фиксированного списка параметров.

- «редалируемый»

Позволяет пользователю вводить текст или выбирать из фиксированного списка параметров.

**parent** - выпадающий родительский контейнер  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Выпадающий родительский контейнер, указанный как окно приложения, панель или объект группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' ). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список стилей и описание свойств (аргументы пары имя-значение) см. В разделе Раскрывающиеся свойства и редактируемые раскрывающиеся свойства.

Пример: «Текст», {«Красный», «Желтый», «Синий»} указывает параметры, представленные в падать.

### Примеры

Создать выпадающий

раскрывающийся список = uidropdown;

---

**Страница 236**

6 Справочник по функциям

Создать редактируемый выпадающий

```
dropdown = uidropdown ( «редактируемый» );
```

---

**Страница 237**

uidropdown

Укажите родительский объект для выпадающего

Укажите окно приложения в качестве родительского объекта для раскрывающегося списка.

```
window = appwindow;
выпадающий = uidropdown (окно);
```

Доступ к раскрывающимся спискам

Создайте раскрывающийся список и укажите значения свойств.

```
dropdown = uidropdown ( 'Text' , { 'Red' , 'Yellow' , 'Blue' , 'Green' } , ...
    «Значение» , «Синий» );
```

6-25

## Страница 238

6 Справочник по функциям

Определите значение данных, связанных с выбранной опцией:

```
vdata = dropdown.ValueData
```

```
vdata =
```

```
3
```

Сделайте шрифт текста, который отображается в выпадающем курсиве:

```
dropdown.FontAngle = 'курсив' ;
```

Код выпадающего обратного вызова ValueChangedFcn

Создайте окно приложения, содержащее оси и выпадающий список. Код обратного вызова для выпадающего так, что когда пользователь выбирает параметр, график на осях меняет цвет.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция plotOptions
% Создание окна приложения и компонентов:
```

6-26

uidropdown

```
window = appwindow ( 'Position' , [100, 100, 350, 275]);

ax = axes ( 'Parent' , окно, ...
    «Единицы» , «пиксели» , ...
    «Положение» , [93, 51, 100, 77]);

% Создать сюжет
x = пробел (-2 * пи, 2 * пи);
y = грех (x);
p = сюжет (топор, x, y);
p.Color = 'Blue' ;

dropdown = uidropdown (окно, ...
    «Местоположение» , [84,204], ...
    «Текст» , { «Красный» , «Желтый» , «Синий» , «Зеленый» }, ...
    «Значение» , «Синий» , ...
    'ValueChangedFcn' , выбор @ (выпадающий список , событие) (выпадающий список, p));
конец

Обратный вызов% Create ValueChangedFcn:
выбор функции (выпадающий, p)
val = dropdown.Value;
p.Color = val;
конец
```

Запустите plotOptions. Выберите опцию из выпадающего списка, чтобы изменить цвет графика.

6-27

Кодирование редактируемого обратного вызова ValueChangedFcn с использованием свойств и свойств данных

Создайте окно приложения, содержащее редактируемый выпадающий список и лампу. Код обратного вызова для раскрывающегося списка, так что когда пользователь выбирает параметр или вводит числовое значение в раскрывающемся списке размер лампы умножается на это значение. Этот пример иллюстрирует как использовать свойства Text и Value, чтобы определить выбор пользователя приложения.

Сохраните следующий код в файл на вашем пути MATLAB.

```

функция lampSize
% Создание окна приложения и компонентов

window = appwindow ( 'Position' , [100, 100, 300, 275]);

лампа = uilamp (окно, ...
    «Местоположение» , [100, 25]);

dropdown = uidropdown (окно, «редактируемый» , ...
    «Местоположение» , [84,204], ...
    'Текст' , { '1' , '2' , '3' , '4' } , ...
    «Значение» , «1» , ...

```

6-28

---

Страница 241

uidropdown

```

'ValueChangedFcn' , @ (выпадающий список, событие) optionSelected (выпадающий список, индикатор));
конец

% Create ValueChangedFcn обратный вызов
функция optionSelected (выпадающий список, лампа)
    val = dropdown.Value;
    s = [20 20];
    переключатель val
        вариант «Вариант 1»
            lamp.Size = s;
        кейс "Вариант 2"
            lamp.Size = s . * [2, 2];
        кейс "Вариант 3"
            lamp.Size = s . * [3, 3];
        кейс "Вариант 4"
            lamp.Size = s . * [4, 4];
    в противном случае
        v = dropdown.Value;
        m = str2num (v);
        lamp.Size = s . * [m, m];
    конец
конец

```

Запустите lampSize и выберите различные варианты из выпадающего списка.

Введите значение в раскрывающемся списке. (Если вы введете большое значение, возможно, вам придется изменить размер Окно приложения, чтобы увидеть лампу.)

---

**Страница 242**

6 Справочник по функциям

Кодирование редактируемого обратного вызова ValueChangedFcn с использованием ValueData и TextData свойства

Создайте окно приложения, содержащее редактируемый выпадающий список и лампу. Код обратного вызова для раскрывающегося списка, так что, когда пользователь выбирает параметр, размер лампы изменяется на размер, подходящий для выбора. Когда пользователь вводит числовое значение в раскрывающемся списке вниз, размер лампы становится этим значением. Этот пример иллюстрирует, как использовать Свойства ValueData и TextData для определения выбора пользователя приложения.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция lampSizer
% Создание окна приложения и компонентов

window = appwindow ( 'Position' , [100, 100, 300, 275]);

лампа = uilamp (окно, ...
    «Местоположение» , [100, 25]);

dropdown = uidropdown (окно, «редактируемый» , ...
    «Местоположение» , [84,204], ...
```

---

**Страница 243**

```

    'Text', { 'small', 'medium', 'large', 'extra large' }, ...
    «Значение», «средний», ...
    'TextData', {10,20,30,40}, ...
    'ValueChangedFcn', @ (выпадающий список, событие) optionSelected (выпадающий список, индикатор));
конец

% Create ValueChangedFcn обратный вызов
функция optionSelected (выпадающий список, лампа)
val = dropdown.ValueData;
if (~ isempty (val))
    % Выбранная пользователем опция
    lamp.Size = val * [1,1];
еще
    % Пользователь ввел значение
    v = dropdown.Value;
    n = str2num (v);
    if (~ isempty (n))
        % Пользователь набрал номер
        lamp.Size = n * [1,1];
еще
        % Если пользователь не набрал номер,
        % использовать размер лампы по умолчанию
        lamp.Size = [20,20];
конец
конец
конец

```

Запустите lampSizer и выберите различные варианты из выпадающего списка.

Введите числовое значение в раскрывающемся списке. (Если вы введете большое значение, вам может понадобиться изменить размер окна приложения, чтобы увидеть лампу.)



Смотрите также

свойства  
Drop Down Свойства | Редактируемые раскрывающиеся свойства

6-32

uieditfield

## uieditfield

Создать текстовое или числовое поле редактирования

### Синтаксис

editfield = uieditfield  
editfield = uieditfield (стиль)  
editfield = uieditfield (родительский)  
editfield = uieditfield (родитель, стиль)  
editfield = uieditfield ( \_\_, Имя, Значение)

### Описание

editfield = uieditfield создает текстовое поле редактирования в новом окне приложения.

editfield = uieditfield (style) создает поле редактирования указанного стиля.

editfield = uieditfield (parent) указывает объект для создания редактирования поле.

editfield = uieditfield (parent, style) создает поле редактирования указанного стиля в указанном родительском объекте.

editfield = uieditfield ( \_\_, Name, Value) определяет свойства поля редактирования используя один или несколько аргументов Имя, Пара значений. Используйте эту опцию с любым входом комбинации аргументов в предыдущих синтаксисах.

### Входные аргументы

**style** - Тип поля редактирования  
«текст» (по умолчанию) | «Числовой»

Тип поля редактирования, указанный как один из следующих:

- «текст»

По умолчанию в полях редактирования текста текст отсутствует.

- «числовой»

По умолчанию в числовых полях редактирования отображается значение 0.

**parent** - редактировать поле родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Редактировать поле родительского контейнера, указанное в качестве окна приложения, панели или объекта группы кнопок.

Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список стилей и описание свойств (аргументы пары имя-значение) см. В разделе Редактировать свойства поля и числовые Редактировать свойства поля.

Пример: «Местоположение», [150,150] указывает координаты из левого нижнего угла родительского контейнера в нижний левый угол поля редактирования (в пикселях).

Примеры

Создать текстовое поле редактирования

Создайте текстовое поле редактирования, стиль по умолчанию для поля редактирования.

editfield = uieditfield;

Создать числовое поле редактирования

Создайте числовое поле редактирования, указав стиль как числовой.

```
editfield = uieditfield ( 'numeric' );
```

6-35

---

## Страница 248

6 Справочник по функциям

Укажите родительский объект для числового поля редактирования

Укажите окно приложения в качестве родительского объекта для числового поля редактирования.

```
window = appwindow;  
editfield = uieditfield (window, 'numeric' );
```

Доступ к числовым значениям поля редактирования

Создайте числовое поле редактирования.

```
editfield = uieditfield ( 'numeric' );
```

Определите пределы:

```
limit = editfield.Limits
```

```
пределы =
```

```
Инф
```

Возвращенные значения указывают на отсутствие ограничений.

6-36

## Страница 249

uieditfield

Укажите пределы от 0 до 100, исключая.

```
editfield.Limits = [0,100];
```

Создайте числовое поле редактирования и укажите предельную включенность

Создайте числовое поле редактирования, которое позволяет пользователям вводить значение больше -5 и меньше чем или равно 10.

```
editfield = uieditfield ( 'числовой' , ...
    «Пределы» , [-5, 10], ...
    «LowerLimitInclusivity» , ложь, ...
    «UpperLimitInclusivity» , правда, ...
    «Значение» , 5);
```

Запустите команду. Если вы введете значение в числовое поле редактирования, которое выходит за пределы, MATLAB автоматически отображает сообщение с указанием проблемы и восстанавливает значение к предыдущему действительному значению.

Создайте числовое поле редактирования, которое отображает значения, используя ровно два десятичных знака

Создайте числовое поле для редактирования, которое позволяет пользователям вводить любое значение, но всегда отображает значение с использованием ровно двух десятичных знаков. Знайте, что MATLAB хранит точное значение, которое пользователь вводит.

```
editfield = uieditfield ( 'числовой' , ...
    'ValueDisplayFormat' , '%.2f' );
```

Введите 5.555 в числовое поле редактирования, а затем щелкните вдали от поля редактирования. Редактирование поле отображает 5.55.

MATLAB сохраняет первоначальное значение 5,555.

Нажмите в поле редактирования, оно отображает первоначально введенное значение.

Кодирование обратного вызова ValueChangedFcn для поля редактирования текста

Запишите обратный вызов ValueChangedFcn для поля редактирования текста, чтобы пользователь изменяет текст в поле редактирования, метка обновляется в соответствии с этим текстом.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция textValue
% Создание окна приложения и компонентов
```

6-37

## Страница 250

6 Справочник по функциям

```
window = appwindow ( 'Position' , [100,100,350,275]);
```

```
label = uilabel (окно, ...
    «Местоположение» , [130, 100], ...
    «Размер» , [100,20]);
```

```
textfield = uieditfield (окно, ...
    «Местоположение» , [100,175], ...
    'ValueChangedFcn' , @ (текстовое поле, событие) textChanged (текстовое поле, метка));
конец
```

функция textChanged (текстовое поле, метка)

```
label.Text = textfield.Value;
```

конец

Запустите textValue и введите Velocity в поле редактирования. Нажмите за пределами поля редактирования, чтобы вызвать обратный вызов.

Кодирование обратного вызова ValueChangedFcn для числового поля редактирования

Кодировать обратный вызов ValueChangedFcn для числового поля редактирования так, чтобы, когда пользователь изменяет значение в поле редактирования, ползунок обновляется в соответствии с этим значением.

6-38

## Страница 251

uieditfield

Сохраните следующий код в файл на вашем пути MATLAB.

Функция numericEditFieldValue

```
% Создание окна приложения и компонентов
```

```
window = appwindow ( 'Position' , [100,100,350,275]);
```

```
слайдер = uislider (окно, ...
```

```
«Местоположение» , [90, 220]);
```

```
числовое поле = uieditfield (окно, 'числовое' , ...
```

```
«Местоположение» , [100,140], ...
```

```
'ValueChangedFcn' , @ (числовое поле, событие) numberChanged (числовое поле, ползунок));
```

конец

```
% Create ValueChangedFcn обратный вызов
```

```
функция numberChanged (числовое поле, ползунок)
```

```
slider.Value = числовое поле.Value;
```

конец

Запустите numericEditFieldValue.

Введите значение от 0 до 100 в поле редактирования чисел и щелкните за пределами поля. ползунок перемещается, чтобы указать числовое значение поля редактирования.

Смотрите также

- функции
- uitextarea
- свойства
- Изменить свойства поля | Числовые свойства поля редактирования

uigauge

Создание кругового, линейного, девяносто-градусного или полукруглого датчика

## Синтаксис

```
датчик = uigauge
датчик = uigauge (стиль)
датчик = uigauge (родитель)
датчик = uigauge (родитель, стиль)
датчик = uigauge (____, имя, значение)
```

## Описание

`gauge = uigauge` создает круговой датчик в новом окне приложения.

`gauge = uigauge (стиль)` определяет стиль датчика.

`gauge = uigauge (parent)` указывает объект, в котором необходимо создать датчик.

`gauge = uigauge (parent, style)` создает калибровку указанного стиля в указанный родительский объект.

`gauge = uigauge (____, Name, Value)` определяет свойства датчика, используя один или больше имен, значения пары аргументов. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**Стиль** - Тип датчика  
круговой (по умолчанию) | «линейный» | девяносто градусов | «Полукруглая»

Тип датчика, указанный как одно из следующих значений:

Стиль	Visual по умолчанию	Описание
«Круговым»		Компонент для отображения значение за круговой промежуток. Это всегда считает с 225 градусов по часовой стрелке направление до 315 градусов.
«Линейный»		Компонент для отображения значение за линейный промежуток.
'Ninetydegree'		Компонент для отображения значение более 90 градусов.
«Полукруглая»		Компонент для отображения значение над полукруглым пяди. Угол развертки фиксируется на 180 градусов.
<b>parent</b> - Датчик родительского контейнера объект окна приложения (по умолчанию)   объект панели   объект группы кнопок		
Измерить родительский контейнер, указанный как окно приложения, панель или объект группы кнопок.		

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список стилей и описание свойств (аргументы пары имя-значение) см. В разделе Калибровочные свойства, полукруглые калибровочные свойства, калибровочные свойства девяносто градусов и Свойства линейного датчика.

Пример: «Значение», 150 указывает, что стрелка датчика указывает на 150.

6-42

---

Страница 255

uigauge

## Примеры

Создать круговой датчик

```
датчик = uigauge;
```

Создать линейный датчик

```
датчик = uigauge ( «линейный» );
```

6-43

---

Страница 256

6 Справочник по функциям



Укажите родительский объект для датчика

Укажите окно приложения в качестве родительского объекта для линейного датчика.

```
window = appwindow;  
датчик = uigauge (окно, «линейный» );
```

Access Gauge Property Values

Создайте круговой датчик:

```
датчик = uigauge;
```

Измените основные отметки, укажите соответствующие метки и удалите второстепенные отметки.

```
gauge.MajorTicks = [0: 10: 100];  
gauge.MajorTickLabels = { '0', '10', '20', '30', '40', '50', '60', '70', '80', '90', '100' };  
gauge.MinorTicks = [];
```

Определить размер датчика:

```
size = gauge.Size
```

6-44

---

Страница 257

uigauge

размер =

120 120

Создайте круговую шкалу с указанием цветов шкалы и пределов цвета шкалы

```
gauge = uigauge ( 'ScaleColors' , { 'yellow' , 'red' } , ...  
                  'ScaleColorLimits' , [60,80; 80100]);
```

6-45

Смотрите также

свойства  
Gauge Properties | Линейный Калибр Свойства | Девяносто-градусный калибр Свойства |  
Свойства полукруглого датчика

6-46

# uiknob

Создание ручки или компонента приложения дискретной ручки

## Синтаксис

```
ручка = uiknob
knob = uiknob (стиль)
knob = uiknob (родитель)
knob = uiknob (родитель, стиль)
ручка = uiknob (__, имя, значение)
```

## Описание

knob = uiknob создает ручку в новом окне приложения.

knob = uiknob (стиль) определяет стиль регулятора.

knob = uiknob (parent) указывает объект, в котором нужно создать кнопку.

knob = uiknob (parent, style) создает ручку указанного стиля в указанном родительский объект.

knob = uiknob (\_\_, Name, Value) определяет свойства ручки, используя один или несколько Имя, Значение пары аргументов. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**стиль** - тип ручки  
«непрерывный» (по умолчанию) | «Дискретное»  
Тип ручки, указанный как одно из следующих значений:

6-47

Стиль	Visual по умолчанию
«Непрерывной»	
«Дискретное»	

**parent** - Ручка родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Ручка родительского контейнера, указанная в качестве окна приложения, панели или объекта группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список стилей и описание свойств см. В разделе Свойства ручки и «Дискретный». Ручка Свойства.

Пример: «Включено», false указывает, что ручка отображается затемненной и не в рабочем состоянии.

## Примеры

Создать ручку по умолчанию

Создайте непрерывную ручку, стиль ручки по умолчанию.

6-48

---

Страница 261

uiknob

```
ручка = uiknob;
```

Создать Дискретную Ручку

Создайте дискретную ручку, указав стиль как дискретный.

```
кноб = uiknob ( «дискретный» );
```

---

Страница 262

6 Справочник по функциям

Укажите родительский объект для дискретной ручки

Укажите окно приложения в качестве родительского объекта для отдельной ручки.

```
window = appwindow;  
knob = uiknob (окно, «дискретный»);
```

Доступ к значениям свойства Continuous Knob

Создать непрерывную ручку.

```
ручка = uiknob;
```

Определите пределы ручки.

```
limit = knob.Limits
```

```
пределы =
```

```
0 100
```

Измените пределы и значение.

6-50

---

Страница 263

uiknob

```
knob.Limits = [-10,10];  
knob.Value = 5;
```

Доступ к значениям свойства дискретной ручки

Создайте дискретную ручку.

```
knob = uiknob ( «дискретный» );
```

Измените состояния ручки. Установите данные, связанные с состояниями ручки, для отражения температура в градусах Фаренгейта.

```
knob.Text = { 'Cold' , 'Warm' , 'Hot' };  
knob.TextData = {32, 80, 212};
```

6-51

---

## Страница 264

6 Справочник по функциям

Получить значение данных (температуру), связанное с текущим значением регулятора.

```
градусы = ручка.ValueData
```

```
градусы =
```

```
32
```

Кодирование обратного вызова ValueChangedFcn для дискретной ручки

Создайте окно приложения, содержащее дискретную ручку и текстовое поле. Код

Обратный вызов ValueChangedFcn такой, что когда пользователь меняет положение ручки, текст поле обновлено, чтобы отразить выбор пользователя.

Скопируйте и вставьте следующий код в файл с именем displayknobvalue.m на вашем MATLAB путь.

```
функция displayKnobValue  
% Создание окна приложения и компонентов
```

```
window = appwindow ( 'Position' , [100, 100, 283, 275]);

textfield = uieditfield (окно, 'текст' , ...
```

6-52

uiknob

```
«Местоположение» , [69, 82]);

knob = uiknob (окно, 'дискретный' , ...
«Местоположение» , [89, 142], ...
'ValueChangedFcn' , @ (ручка, событие) knobTurned (ручка, текстовое поле));
конец

функция knobTurned (ручка, текстовое поле)
    textfield.Value = knob.Value;
конец

Запустите displayKnobValue, а затем поверните ручку. Когда вы отпустите кнопку мыши,
поле редактирования обновляется, чтобы отразить значение регулятора.
```

Кодирование обратного вызова ValueChangedFcn для непрерывной ручки

Создайте окно приложения, содержащее непрерывную ручку и метку. Код

Обратный вызов ValueChangedFcn такой, что, когда пользователь меняет положение ручки, метка обновляется, чтобы отразить выбор пользователя.

Скопируйте и вставьте следующий код в файл с именем showknobvalue.m на вашем MATLAB путь.

```
функция showKnobValue
% Создание окна приложения и компонентов

window = appwindow ( 'Position' , [100, 100, 283, 275]);
```

6-53

6 Справочник по функциям

```
label = uilabel (окно, 'Местоположение' , [218,177], ...
«Текст» , «0» , ...
«Размер» , [50,15]);

knob = uiknob (окно, ...
«Местоположение» , [89, 142], ...
'ValueChangedFcn' , @ (ручка, событие) knobTurned (ручка, метка));
конец
```

```
% Create ValueChangedFcn обратный вызов
```

```
функция knobTurned (ручка, метка)
```

```
num = knob.Value;
```

```
label.Text = num2str (num);
```

```
конец
```

Запустите showKnobValue и поверните ручку. Когда вы отпустите кнопку мыши, ярлык обновлен, чтобы отразить значение ручки.

Кодирование обратного вызова ValueChangingFcn для непрерывной ручки

Создайте окно приложения, содержащее непрерывную ручку и числовое поле редактирования. Код обратный вызов ValueChangingFcn ручки, так что, когда пользователь меняет положение ручки,

6-54

## Страница 267

uiknob

поле редактирования постоянно обновляется, чтобы отразить значение регулятора. MATLAB хранит изменение значения ручки в данных события обратного вызова.

Скопируйте и вставьте следующий код в файл с именем showchangingvalue.m на вашем MATLAB путь.

```
функция showChangingValue
```

```
% Создание окна приложения и компонентов
```

```
window = appwindow ( 'Position' , [100, 100, 350, 275]);
```

```
число поле = uieditfield (окно, 'числовой' , ...
```

```
«Местоположение» , [218,160]);
```

```
knob = uiknob (окно, ...
```

```
«Местоположение» , [89, 142], ...
```

```
'ValueChangingFcn' , @ (ручка, событие) knobTurned (ручка, событие, поле номера));
```

```
конец
```

```
% Create ValueChangingFcn обратный вызов
```

```
функция knobTurned (ручка, событие, поле номера)
```

```
numberfield.Value = event.Value;
```

```
конец
```

Запустите showChangingValue и поверните ручку. Когда вы это сделаете, числовое поле редактирования обновлен, чтобы показать изменяющиеся значения регулятора.



6-55

Страница 268

6 Справочник по функциям

Смотрите также

свойства  
Свойства дискретной ручки | Knob Properties

6-56

Страница 269

UILabel

UILabel

Создать компонент метки

## Синтаксис

```
ярлык = uilabel
label = uilabel (родитель)
label = uilabel (____, имя, значение)
```

## Описание

label = uilabel создает метку компонента с текстом Label.

label = uilabel (parent) указывает объект, в котором нужно создать метку.

label = uilabel (\_\_\_\_, Name, Value) определяет свойства метки, используя один или больше имени, значения пары аргументов. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**parent** - метка родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Пометить родительский контейнер, указанный как окно приложения, панель или объект группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты (' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список и описание свойств (аргументы пары имя-значение) см. В разделе «Метка». Свойства.

6-57

## Страница 270

6 Справочник по функциям

Пример: «Текст», «Размер:» указывает, что метка отображает размер текста .:

**«Текст»** - текст метки  
«Метка» (по умолчанию) | строка | массив ячеек строк

Текст метки, указанный как строка или массив ячеек строк. Используйте массив строк для указать многострочную метку.

MATLAB может корректно отображать форматированный текст, например:

```
text = sprintf( '%s \n %s', 'Строка 1', 'Строка 2' )
label = uilabel ( «Текст», текст, «Размер», [100,100])
```

Однако MATLAB автоматически не интерпретирует и не отображает текст, такой как этот:

```
label = uilabel ( «Текст», «Строка 1 \n Line2», «Размер», [100,150])
```

Пример: «Порог»

Пример: {'Порог' 'Значение'}

Примеры

Создать компонент метки

ярлык = uilabel;

6-58

UILabel

Укажите родительский объект для компонента лампы

Укажите окно приложения в качестве родительского объекта для метки.

```
window = appwindow;  
метка = uilabel (окно);
```

Свойства ярлыка доступа

Создайте метку по умолчанию.

```
ярлык = uilabel;
```

6-59

Измените текст метки и размер шрифта.

```
label.Text = 'Result' ;  
label.FontSize = 14;
```

Метка обрезана, потому что текущий размер недостаточно велик для измененного текста.

Определите текущий размер этикетки.

размер = метка. Размер

6-60

---

Страница 273

UILabel

размер =

31 15

Измените размер этикетки, чтобы приспособить измененный текст.

```
label.Size = [60,20];
```

Смотрите также

свойства  
Свойства ярлыка

6-61

# uilamp

Создать компонент лампы

## Синтаксис

лампа = Uilamp  
лампа = uilamp (родитель)  
лампа = uilamp ( \_\_, имя, значение)

## Описание

lamp = uilamp создает компонент лампы в новом окне приложения.

lamp = uilamp (parent) указывает объект, в котором нужно создать лампу.

lamp = uilamp ( \_\_, Name, Value) определяет свойства лампы, используя один или несколько Имя, Значение пары аргументов. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**parent** - Лампа родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Лампа родительского контейнера, указанная в качестве окна приложения, панели или объекта группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' ). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список и описание свойств см. В разделе «Свойства лампы».

Пример: «Цвет», «красный» указывает, что цвет лампы красный.

---

**Страница 275**

uilamp

## Примеры

Создать компонент лампы

Создайте зеленый компонент лампы.

```
лампа = uilamp;
```

Укажите родительский объект для компонента лампы

Укажите окно приложения в качестве родительского объекта для лампы.

```
window = appwindow;  
лампа = uilamp (окно);
```

Свойства лампы доступа

Создайте лампу по умолчанию.

```
лампа = uilamp;
```

**6-63**

---

**Страница 276****6** Справочник по функциям

Определите текущий цвет лампы.

```
цвет = лампа. цвет
```

```
цвет =
```

```
0 1 0
```

MATLAB возвращает значение RGB для зеленого цвета.

Измените цвет лампы на красный.

```
lamp.Color = 'red' ;
```

6-64

---

Страница 277

uilamp

Измените цвет лампы на синий.

```
лампа. Цвет = [0,0,1];
```

Смотрите также

свойства  
Свойства лампы

# uilibox

Создать компонент списка

## Синтаксис

```
listbox = uilibox
listbox = uilibox (родительский)
listbox = uilibox (__, имя, значение)
```

## Описание

listbox = uilibox создает список в новом окне приложения.

listbox = uilibox (parent) указывает объект, в котором нужно создать список.

listbox = uilibox (\_\_, Name, Value) определяет свойства списка, используя один или более Имя, Аргументы пары Значение. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**parent** - список родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер списка, указанный как окно приложения, панель или объект группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список свойств см. В разделе Свойства списка.

Пример: «Текст», {«Модель 1», «Модель 2», «Модель 3», «Модель 4»} указывает параметры списка, которые видит пользователь, сверху вниз.

## Примеры

Создать список

```
listbox = uilibox;
```



Укажите родительский объект для списка

Укажите окно приложения в качестве родительского объекта для списка.

```
window = appwindow;
listbox = uilistbox (окно);
```

Значения свойств окна списка доступа

Создайте список по умолчанию.

6-67

## Страница 280

6 Справочник по функциям

```
listbox = uilistbox;
```

Определите, допускает ли список несколько вариантов выбора.

```
multi = listBox.MultiSelect
```

```
мульти =
```

```
0
```

MATLAB возвращает ноль, указывая, что множественный выбор не включен.

Включить множественный выбор.

```
listbox.MultiSelect = true;
```

Кодирование обратного вызова ValueChangedFcn для списка для отображения выбора

Создайте окно приложения, содержащее список и поле для редактирования текста. Код обратного вызова для поле списка, так что когда пользователь выбирает опцию списка, MATLAB отображает эту опцию в тексте поле редактирования.

Сохраните следующий код в файл на вашем пути MATLAB.

Функция listBoxSelection

```
% Создание окна приложения и компонентов
```

```
window = appwindow ( 'Position', [100,100,350,275]);
```

```
textfield = uieditfield (окно, ...
```

```
    «Местоположение», [86,90]);
```

```
    textfield.Value = 'Item 1' ;
```

```
lbox = uilistbox (окно, ...
```

```
    «Местоположение», [100, 120], ...
```

```
    'ValueChangedFcn', @ (lbox, событие) selectionChanged (lbox, текстовое поле));
```

```
конец
```

```
% Create ValueChangedFcn обратный вызов
```

```
функция selectionChanged (Ibox, текстовое поле)
textfield.Value = Ibox.Value;
конец
```

Запустите listBoxSelection и выберите вариант из списка. Поле для редактирования текста обновлено чтобы отразить ваш выбор.

6-68

uilibox

Кодирование обратного вызова ValueChangedFcn для списка для отображения данных, связанных с выбором

Создайте окно приложения, содержащее список и числовое поле редактирования. Код обратного вызова для поля списка, так что когда пользователь выбирает опцию списка, MATLAB отображает данные связан с этой опцией в числовом поле редактирования.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция dataSelection
% Создание окна приложения и компонентов

window = appwindow ( 'Position' , [100,100,350,275]);

числовое поле = uieditfield (окно, 'числовое' , ...
    «Местоположение» , [86,90]);

Ibox = uilibox (окно, ...
    «Текст» , { «Замораживание» , «Тепло» , «Горячее» , «Кипение» }, ...
    'TextData' , {0, 25, 40, 100}, ...
    «Местоположение» , [100, 120], ...
```

6-69

```
'ValueChangedFcn' , @ (Ibox, событие) selectionChanged (Ibox, числовое поле);
конец

% Create ValueChangedFcn обратный вызов
функция selectionChanged (Ibox, числовое поле)
```

```
numericfield.Value = lbox.ValueData;  
конец
```

Запустите dataSelection и выберите опцию из списка. Числовое поле редактирования обновлен, чтобы отразить данные, связанные с вашим выбором.

Кодирование обратного вызова ValueChangedFcn для списка с несколькими выборками

Создайте окно приложения, содержащее список и текстовое поле. Код обратного вызова для поле списка, так что когда пользователь выбирает один или несколько параметров списка, MATLAB отображает их варианты в текстовой области.

Сохраните следующий код в файл на вашем пути MATLAB.

функция множественного выбора

6-70

---

Страница 283

uilibox

```
% Создание окна приложения и компонентов
```

```
window = appwindow ( 'Position' , [100,100,350,275]);
```

```
textarea = uitable (окно, ...  
«Местоположение» , [62,90]);
```

```
listbox = uilibox (окно, ...  
«Местоположение» , [100, 150], ...  
«Мультивыбор» , правда, ...  
'ValueChangedFcn' , @ (список, событие) selectionChanged (список, текстовое поле));
```

```
конец
```

```
% Create ValueChangedFcn обратный вызов
```

```
функция selectionChanged (listbox, textarea)
```

```
textarea.Value = listbox.Value;
```

```
конец
```

Запустите multiselect и выберите параметры из списка. Поле для редактирования текста обновлено до отразить ваш выбор.

6-71

---

Страница 284

6 Справочник по функциям

Смотрите также

свойства  
Свойства списка

6-72

---

Страница 285

uiradiobutton

## uiradiobutton

Создать компонент радиокнопки

### Синтаксис

`radiobutton = uiradiobutton (родитель)`  
`radiobutton = uiradiobutton (родитель, имя, значение)`

### Описание

`radiobutton = uiradiobutton (parent)` создает переключатель в пределах указанного группа родительских кнопок.

`radiobutton = uiradiobutton (parent, Name, Value)` указывает переключатель свойства, использующие один или несколько аргументов Name, Value.

### Входные аргументы

**parent** - переключатель родительского контейнера  
объект `matlab.ui.control.ButtonGroup`

Родительский контейнер переключателя, указанный как `matlab.ui.control.ButtonGroup` объект.

#### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Пример: «Текст», «Французский» указывает, что текст, французский, отображается рядом с радио кнопка.

6-73

### Примеры

Создание радиокнопок внутри группы кнопок и доступ к значениям свойств

Чтобы создать переключатель, сначала создайте окно приложения и объект группы кнопок.

```
window = appwindow;  
bg = matlab.ui.control.ButtonGroup ( «Родитель» , окно);
```

Создайте три переключателя и укажите расположение каждого из них.

```
rb1 = uiradiobutton (bg, «Местоположение» , [10,60]);  
rb2 = uiradiobutton (bg, «Местоположение» , [10,38]);  
rb3 = uiradiobutton (bg, «Местоположение» , [10,16]);
```

Изменить текст, связанный с каждой радиокнопкой.

```
rb1.Text = 'English' ;  
rb2.Text = 'Французский' ;  
rb3.Text = 'German' ;
```

6-74

uiradiobutton

Измените выбор переключателя на немецкий.

```
rb3.Value = true;
```

6-75

Определите название шрифта немецкого текста переключателя.

```
font = rb3.FontName
```

шрифт =

Helvetica

Смотрите также

[uitogglebutton](#)

6-76

## uislider

Создать компонент слайдера

### Синтаксис

```
слайдер = uislider
слайдер = uislider (родитель)
слайдер = uislider (__, имя, значение)
```

### Описание

slider = uislider создает слайдер в новом окне приложения.

slider = uislider (parent) указывает объект, в котором нужно создать слайдер.

`slider = uislider(Name, Value)` создает слайдер, используя один или несколько аргументов комбинации в предыдущих синтаксисах.

## Входные аргументы

**parent** - ползунок родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер ползунка, указанный как окно приложения, панель или объект группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' ). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список и описание свойств (пары имя-значение) см. В разделе «Свойства ползунка».

Пример: «Limits», [0,50] задает минимальное значение ползунка как 0 и максимальное значение ползунка как 50.

## Примеры

Создать слайдер

слайдер = uislider;

Укажите родительский объект для слайдера

```
window = appwindow;  
слайдер = uislider (окно);
```

Доступ к значениям свойств слайдера

Создайте ползунок по умолчанию.

```
слайдер = uislider;
```

Определите текущие пределы слайдера.

```
limit = slider.Limits
```

пределы =



---

**Страница 291**

uislider

0 100

Измените пределы ползунка и установите значение 35.

```
slider.Limits = [-50,50];
slider.Value = 35;
```

Кодирование обратного вызова ValueChangedFcn для ползунка

Кодировать обратный вызов ValueChangedFcn для ползунка для управления иглой круглого калибра.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция sliderValue
% Создание окна приложения и компонентов

window = appwindow ( 'Position' , [100,100,350,275]);

gauge = uigauge (окно, «Местоположение» , [100,100]);

слайдер = uislider (окно, ...
    «Местоположение» , [100, 75], ...
```

6-79

---

**Страница 292**

6 Справочник по функциям

```
'ValueChangedFcn' , @ (слайдер, событие) sliderMoved (слайдер, индикатор));
```

```
конец
```

```
% Create ValueChangedFcn обратный вызов
```

```
функция sliderMoved (слайдер, датчик)
```

```
gauge.Value = slider.Value;
```

```
конец
```

Запустите sliderValue, а затем переместите ползунок. Когда вы отпустите кнопку мыши, круговая стрелка перемещается к соответствующему значению на датчике.

Кодирование обратного вызова ValueChangingFcn для ползунка

Закодируйте обратный вызов ValueChangingFcn для ползунка для управления иглой кругового калибра.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция sliderChanging
% Создание окна приложения и компонентов

window = appwindow ( 'Position' , [100,100,350,275]);
```

6-80

---

## Страница 293

uislider

```
gauge = uigauge (окно, «Местоположение» , [100,100]);

слайдер = uislider (окно, ...
    «Местоположение» , [100, 75], ...
    'ValueChangingFcn' , @ (слайдер, событие) sliderMoving (событие, индикатор));

конец

% Create ValueChangingFcn обратный вызов
функция sliderMoving (событие, датчик)
gauge.Value = event.Value;
конец
```

Запустите sliderChanging, а затем переместите ползунок. Когда вы перемещаете ползунок, круговой стрелка датчика перемещается, отражая значение ползунка.

Смотрите также

## uiswitch

Создание ползункового переключателя, тумблера или компонента тумблера

### Синтаксис

контроль = выключатель  
control = uiswitch (стиль)  
control = uiswitch (родительский)  
control = uiswitch (родитель, стиль)  
control = uiswitch (\_\_\_, Name, Value)

### Описание

control = uiswitch создает компонент ползункового переключателя в новом окне приложения.

control = uiswitch (style) создает переключатель указанного стиля.

control = uiswitch (parent) указывает объект, в котором нужно создать переключатель.

control = uiswitch (parent, style) создает переключатель указанного стиля в указанный родительский объект.

control = uiswitch (\_\_\_, Name, Value) определяет свойства переключателя, используя один или больше имени, значения пары аргументов. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

### Входные аргументы

**Стиль** - Тип переключателя  
«слайдер» (по умолчанию) | рокер | «Тумблер»

Тип переключателя, указанный как один из следующих стилей:

Стиль	Visual по умолчанию
«Слайдер»	

Стиль	Visual по умолчанию
«Коромысло»	

«Тумблер»

**parent** - переключить родительский контейнер  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Переключить родительский контейнер, указанный как окно приложения, панель или объект группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' '). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Для специфичного для стиля списка и описания свойств, см. Switch Properties, Rocker  
Свойства переключателя и Свойства переключателя.

Пример: «Текст», {'1', '2'} указывает текст, который отображается для двух  
переключать состояния.

## Примеры

Создать слайдер

Создайте ползунок, стиль по умолчанию для переключателя.

ползунковый переключатель = uiswitch;

6-83

Создать тумблер

Создайте тумблер, указав стиль как тумблер.

toggleswitch = uiswitch ( 'toggle' );

6-84

uiswitch

Укажите родительский объект для переключателя

Укажите окно приложения в качестве родительского объекта для клавишного переключателя.

```
window = appwindow;
rockerswitch = uiswitch (window, 'rocker' );
```

Значения свойств переключателя доступа

Создайте тумблер.

```
rockerswitch = uiswitch ( 'рокер' );
```

Изменить текст переключателя.

```
rockerswitch.Text = { 'Stop', 'Start' };
```

Определите текущее значение переключателя.

```
val = rockerswitch.Value

val =
```

6-85

MATLAB возвращает ноль, указывая, что переключатель направлен вниз в положение Stop. (Если вы измените значение свойства «Ориентация» на «горизонтальный», переключатель будет указывать на оставается, когда свойство Value равно нулю.)

Кодирование обратного вызова ValueChangedFcn для клавишного переключателя

Создайте окно приложения, содержащее лампу и тумблер. Код обратного вызова для тумблера, так что когда пользователь нажимает на переключатель, лампа меняет цвет.

Сохраните следующий код в файл на вашем пути MATLAB.

```
функция lampSwitch
% Создание окна приложения и компонентов

window = appwindow ( 'Position' , [100,100,350,275]);

лампа = uilamp (окно, ...
    «Местоположение» , [150,75]);

rockerswitch = uiswitch (window, 'toggle' , ...
    «Местоположение» , [150, 160], ...
    «Ценность» , правда, ...
    'ValueChangedFcn' , @ (rockerswitch, event) switchMoved (rockerswitch, lamp));

конец

% Create ValueChangedFcn обратный вызов
функция switchMoved (тумблер, лампа)
переключатель rockerswitch.Value
    случай 1
        lamp.Color = 'green' ;
    случай 0
        lamp.Color = 'black' ;
конец
конец
```

Запустите lampSwitch, а затем переместите переключатель вверх и вниз.

Смотрите также

свойства  
Rocker Switch Properties | Переключатель Свойства | Свойства тумблера

# uitextarea

Создать компонент текстовой области

## Синтаксис

```
textarea = uitextarea
textarea = uitextarea (родитель)
textarea = uitextarea ( __, имя, значение)
```

## Описание

textarea = uitextarea создает текстовую область в новом окне приложения.

textarea = uitextarea (parent) указывает объект, в котором нужно создать текстовую область.

textarea = uitextarea ( \_\_, Name, Value) определяет свойства текстовой области, используя один или более Имя, Аргументы пары Значение. Используйте эту опцию с любым входным аргументом комбинации в предыдущих синтаксисах.

## Входные аргументы

**parent** - родительский контейнер текстовой области  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Родительский контейнер текстовой области, указанный как окно приложения, панель или объект группы кнопок.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов Name, Value. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' ). Вы можете указать несколько аргументов имя и значение в любом порядке:

Name1, Value1, ..., NameN, ValueN.

Список и описание свойств см. В разделе «Свойства текстовой области».

Пример: «Редактируемый», false указывает, что пользователь не может изменить текст текстовой области.

## Примеры

Создать текстовую область

```
textarea = uitablearea;
```

Укажите родительский объект для текстовой области

Укажите окно приложения в качестве родительского объекта для текстовой области.

```
window = appwindow;  
textarea = uitablearea (окно);
```

Доступ к свойствам текстовой области

Создайте заполненную текстовую область.

```
textareafield = uitablearea ( ...  
                             «Значение» , { «Имя, Фамилия» ; ...  
                             «Адрес 1» ; «Адрес 2» ; «Город, штат, почтовый индекс» });
```

6-89



```
текст.  
  
Определите текущий размер текстовой области.  
  
size = textareafld.Size  
  
размер =  
  
    150 50  
  
Увеличьте размер области текста, чтобы вместить весь текст.  
  
textareafld.Size = [150,67];
```

```
Кодирование ValueChangedFcn для текстовой области  
  
Создайте окно приложения, содержащее текстовую область и две метки. Код обратного вызова для текстовая область, так что, когда пользователь вводит текст и щелкает в области текста, метка обновлен, чтобы поблагодарить пользователя за ввод. Если пользователь удаляет текст, текст спасибо устранен.  
  
функциональные комментарии  
% Создание окна приложения и компонентов  
  
window = appwindow;  
  
label1 = uilabel (окно, ...  
    «Размер» , [100,15], ...  
    «Местоположение» , [100,164], ...  
    «Текст» , «Введите комментарии» );  
  
label2 = uilabel (окно, ...  
    «Размер» , [175,15], ...  
    «Местоположение» , [100,75], ...  
    «Текст» , «» );
```

## Страница 304

6 Справочник по функциям

```

textarea = uitextarea (окно, ...
    «Местоположение», [100,100], ...
    'ValueChangedFcn', @(textarea, event) textEntered (textarea, label2));

% Create ValueChangedFcn обратный вызов
функция textEntered (textarea, label2)
    val = textarea.Value;
    label2.Text = " ;
    % Проверьте каждый элемент массива ячеек текстовой области на наличие текста
    для k = 1: длина (val)
        if (~ isempty (val {k}))
            label2.Text = 'Спасибо за ваши комментарии!' ;
            перерыв ;
        конец
    конец
конец
конец

```

Запустите комментарии и введите текст в поле текстовой области. Нажмите вдали от текстовой области, чтобы вызвать обратный вызов.

6-92

## Страница 305

uitextarea

Смотрите также

функции  
uieditfield

свойства  
Свойства текстовой области

6-93

---

## Страница 306

6 Справочник по функциям

## uitogglebutton

Создать компонент кнопки переключения

### Синтаксис

```
togglebutton = uitogglebutton (родитель)  
togglebutton = uitogglebutton (родитель, имя, значение)
```

### Описание

`togglebutton = uitogglebutton (parent)` создает кнопку переключения внутри указанная группа родительских кнопок.

`togglebutton = uitogglebutton (parent, Name, Value)` указывает кнопку переключения свойства, использующие один или несколько аргументов `Name, Value`.

### Входные аргументы

**parent** - кнопка переключения родительского контейнера  
объект окна приложения (по умолчанию) | объект панели | объект группы кнопок

Переключить родительский контейнер кнопки, указанный как окно приложения, панель или группа кнопок объект.

### Аргументы пары имя-значение

Укажите необязательные разделенные запятыми пары аргументов `Name, Value`. Имя это Имя аргумента и значение - это соответствующее значение. Имя должно появляться внутри одного цитаты ( ' ). Вы можете указать несколько аргументов имя и значение в любом порядке:

`Name1, Value1, ..., NameN, ValueN`.

Пример: «Текст», «Французский» указывает, что текст, французский, отображается на переключателе кнопка.

## Примеры

Создание кнопок переключения в группе кнопок и доступ к значениям свойств

Чтобы создать кнопку переключения, сначала создайте окно приложения и объект группы кнопок.

```
window = appwindow;  
bg = matlab.ui.control.ButtonGroup ( «Родитель» , окно);
```

Создайте три кнопки переключения и укажите расположение каждой из них.

```
tb1 = uitogglebutton (bg, «Location» , [10,60]);  
tb2 = uitogglebutton (bg, «Местоположение» , [10,38]);  
tb3 = uitogglebutton (bg, «Location» , [10,16]);
```

Изменить текст, связанный с каждой кнопкой переключения.

```
tb1.Text = 'English' ;  
tb2.Text = 'French' ;  
tb3.Text = 'German' ;
```

Измените выбор кнопки переключения на немецкий.

```
tb3.Value = true;
```

6-96

---

Страница 309

uitogglebutton

Определите имя шрифта текста немецкого переключателя.

```
font = tb3.FontName
```

```
шрифт =
```

```
Helvetica
```

Смотрите также

[uicontrol](#)

6-97

---

Страница 310

6-98