

















# Сведения о курсорах

Windows предоставляет набор стандартных **курсов** , которые могут использоваться приложениями. В WinUser.h определены следующие идентификаторы курсоров:







 Развернуть таблицу






Значение	Значение
IDC_ARROW MAKEINTRESOURCE(32512)	 Обычный выбор
IDC_IBEAM MAKEINTRESOURCE(32513)	 Выбор текста
IDC_WAIT MAKEINTRESOURCE(32514)	 Занят
IDC_CROSS MAKEINTRESOURCE(32515)	 Выбор точности
IDC_UPARROW MAKEINTRESOURCE(32516)	 Альтернативный выбор
IDC_SIZEWSE MAKEINTRESOURCE(32642)	 Размер диагонали 1
IDC_SIZENSW MAKEINTRESOURCE(32643)	 Размер диагонали 2
IDC_SIZEWE MAKEINTRESOURCE(32644)	 Горизонтальное изменение размера
IDC_SIZENS MAKEINTRESOURCE(32645)	 Вертикальное изменение размера
IDC_SIZEALL MAKEINTRESOURCE(32646)	 Переместить
IDC_NO MAKEINTRESOURCE(32648)	 Недоступен

Значение	Значение
IDC_HAND MAKEINTRESOURCE(32649)	 Выбор ссылки
IDC_APPSTARTING MAKEINTRESOURCE(32650)	 Работа в фоновом режиме
IDC_HELP MAKEINTRESOURCE(32651)	 Выбор справки
IDC_PIN MAKEINTRESOURCE(32671)	 Выбор расположения
IDC_PERSON MAKEINTRESOURCE(32672)	 Выбор пользователя

Также доступно несколько дополнительных курсоров, которые не имеют идентификаторов, определенных в WinUser.h (или считаются устаревшими):

Развернуть таблицу

Значение	Значение
MAKEINTRESOURCE(32631)	 Курсор пера.
MAKEINTRESOURCE(32652)	 Курсор прокрутки со стрелками, указывающими на север и юг.
MAKEINTRESOURCE(32653)	 Курсор прокрутки со стрелками, указывающими на запад и восток.
MAKEINTRESOURCE(32654)	 Курсор прокрутки со стрелками, указывающими на север, юг, восток и запад.
MAKEINTRESOURCE(32655)	 Курсор прокрутки со стрелкой, указывающей на север.
MAKEINTRESOURCE(32656)	 Курсор прокрутки со стрелкой, указывающей на юг.
MAKEINTRESOURCE(32657)	 Курсор прокрутки со стрелкой, указывающей на запад.
MAKEINTRESOURCE(32658)	 Курсор прокрутки со стрелкой, указывающей на восток.

Значение	Значение
MAKEINTRESOURCE(32659)	 Курсор прокрутки со стрелками, указывающими на север и запад.
MAKEINTRESOURCE(32660)	 Курсор прокрутки со стрелками, указывающими на север и восток.
MAKEINTRESOURCE(32661)	 Курсор прокрутки со стрелками, указывающими на юг и запад.
MAKEINTRESOURCE(32662)	 Курсор прокрутки со стрелками, указывающими на юг и восток.
MAKEINTRESOURCE(32663)	 Курсор на компакт-диске со стрелками.

Дополнительные сведения об использовании стандартных курсоров см. в руководстве.

Каждый стандартный курсор имеет соответствующее изображение по умолчанию, связанное с ним. Пользователь или приложение могут заменить изображение по умолчанию, связанное с любым стандартным курсором в любое время. Приложение заменяет образ по умолчанию с помощью [функции SetSystemCursor](#).

Приложение может использовать [функцию GetIconInfo](#) для получения текущего изображения для курсора и может нарисовать курсор с помощью [функции DrawIconEx](#).

Пользовательские курсоры предназначены для использования в определенном приложении и могут быть любым проектом, определяемого разработчиком. На следующем рисунке показаны несколько пользовательских курсоров.



Курсоры могут быть монохромными или цветными, а также статическими или анимированными. Тип курсора, используемого в определенной компьютерной системе, зависит от отображения системы. Старые экраны, такие как VGA, не поддерживают цвет или анимированные курсоры. Новые экраны, драйверы отображения которых используют подсистему точечных карт (DIB) независимо от устройства, поддерживают их.

Курсоры и значки похожи и могут использоваться взаимозаменяемо во многих ситуациях. Единственное различие между ними заключается в том, что изображение, указанное как курсор, должно быть в формате, который может поддерживать дисплей. Например, курсор должен быть монохромным для отображения VGA.

В этом обзоре содержатся сведения о следующих разделах:

- [Горячее место](#)
- [Мышь и курсор](#)
- [Создание курсора](#)
- [Расположение курсора и внешний вид](#)
- [Ограничение курсора](#)
- [Уничтожение курсора](#)
- [Дублирование курсора](#)
- [Курсор класса Window](#)

## Горячее место

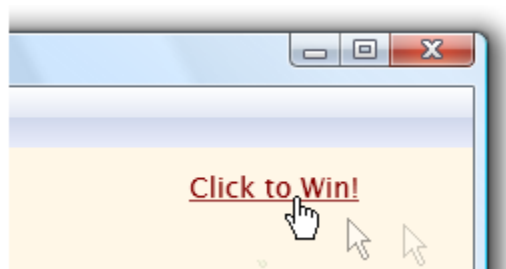
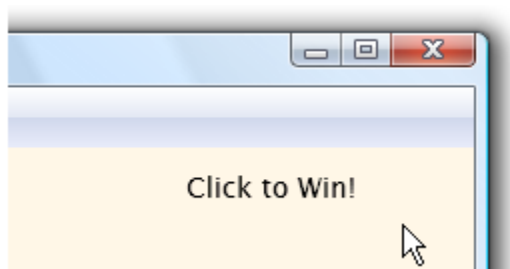
В курсоре пиксель, называемый *горячей точкой*, помечает точное расположение экрана, на которое влияет событие мыши, например нажатие кнопки мыши. Как правило, горячая точка является фокусом курсора. Система отслеживает и распознает эту точку как позицию курсора. Например, типичные горячие точки — это пиксель на кончике курсора со стрелками и пикселя в середине перекрестного курсора. На следующих изображениях показаны два курсора из программы рисования, в которых горячие точки связаны с кончиком кисти и перекрестной чертой краски.



При возникновении события ввода мыши драйвер мыши преобразует событие в соответствующее сообщение мыши, включающее координаты горячей точки. Система отправляет сообщение мыши в окно, содержащее горячее место или в окно, которое записывает входные данные мыши. Дополнительные сведения см. в разделе "[Ввод мыши](#)".

## Мышь и курсор

Система отражает движение мыши, перемещая курсор на экране соответствующим образом. При перемещении курсора по разным частям окон или в разные окна система (или приложение) изменяет внешний вид курсора. Например, когда курсор пересекает гиперссылку, система изменяет курсор с стрелки на руку.



Если у системы нет мыши, система отображает и перемещает курсор, только если пользователь выбирает определенные системные команды, например те, которые используются для размера или перемещения окна. Чтобы предоставить пользователю метод отображения и перемещения курсора, если мышь недоступна, приложение может использовать функции курсора для имитации перемещения мыши. Учитывая эту возможность моделирования, пользователь может использовать клавиши со стрелками для перемещения курсора.

## Создание курсора

Так как стандартные курсоры предопределяются, их не нужно создавать. Чтобы использовать стандартный курсор, приложение извлекает дескриптор курсора с помощью [функции LoadCursor](#) или [LoadImage](#). *Дескриптор* курсора — это уникальное значение типа `HCURSOR`, определяющего стандартный или настраиваемый курсор.

Для создания пользовательского курсора для приложения обычно используется графическое приложение и включается курсор в качестве ресурса в файл определения ресурсов приложения. Во время выполнения вызовите [LoadCursor](#), чтобы получить дескриптор курсора. Ресурсы курсора содержат данные для нескольких различных устройств отображения. Функция [LoadCursor](#) автоматически выбирает наиболее подходящие данные для текущего устройства отображения. Загрузка курсора непосредственно из `.CUR` или `.ANI`-файл используйте функцию [LoadCursorFromFile](#).

Вы также можете создать пользовательский курсор во время выполнения с помощью [функции CreateIconIndirect](#), которая создает курсор на основе содержимого [структуры ICONINFO](#). Функция [GetIconInfo](#) заполняет эту структуру координатами горячей точки и сведениями о связанной маске и цвете.

Приложения должны реализовывать пользовательские курсоры в качестве ресурсов и использовать [LoadCursor](#), [LoadCursorFromFile](#) или [LoadImage](#), а не создавать курсор во время выполнения. Использование ресурсов курсоров позволяет избежать зависимости устройств, упрощает локализацию и позволяет приложениям совместно использовать макеты курсоров.

Функция [CreateIconFromResourceEx](#) позволяет приложению создавать значки и курсоры на основе данных ресурсов. [CreateIconFromResourceEx](#) создает курсор на основе двоичных данных ресурсов из других исполняемых файлов .exe или БИБЛИОТЕК DLL. Перед ним должны предшествовать вызовы [функции LookupIconIdFromDirectoryEx](#), а также несколько функций ресурсов. [LookupIconIdFromDirectoryEx](#) определяет наиболее подходящие данные курсора для текущего устройства отображения. Дополнительные сведения о функциях ресурсов см. в разделе "[Ресурсы](#)".

## Расположение курсора и внешний вид

Система автоматически отображает курсор мыши и обновляет его положение на экране. Вы можете получить текущие координаты экрана курсора и переместить курсор в любое расположение на экране с помощью [функций GetCursorPos](#) и [SetCursorPos](#) соответственно.

Вы также можете получить дескриптор к текущему курсору с помощью [функции GetCursor](#) и задать курсор с помощью [функции SetCursor](#). После вызова [SetCursor](#) внешний вид курсора не изменяется до тех пор, пока мышь не перемещается, курсор явно устанавливается на другой курсор или выполняется системная команда.

Когда пользователь перемещает мышь, система перерисовывает курсор в новом расположении. Система автоматически перерисовывает макет курсора, связанный с окном, на которое указывает курсор.

Вы можете скрыть и повторно воспроизвести курсор, не изменив конструктор курсора, с помощью [функции ShowCursor](#). Эта функция использует внутренний счетчик для определения времени скрытия или отображения курсора. Попытка показать курсор увеличивает счетчик; попытка скрыть курсор уменьшает счетчик. Курсор отображается только в том случае, если этот счетчик больше или равен нулю.

Функция [GetCursorInfo](#) получает следующие сведения для глобального курсора: скрыт ли курсор или отображается, дескриптор курсора и координаты курсора.

## Ограничение курсора

Курсор можно ограничить прямоугольной областью на экране с помощью [функции ClipCursor](#). Это полезно, если пользователь должен реагировать на определенное событие в ограниченной области прямоугольника. Например, можно использовать [ClipCursor](#), чтобы ограничить курсор модальным диалоговым окном, предотвращая взаимодействие пользователя с другими окнами, пока диалоговое окно не будет закрыто.

Функция [GetClipCursor](#) извлекает координаты экрана прямоугольной области, к которой курсор временно ограничен. При необходимости ограничить курсор можно также использовать эту функцию для сохранения координат исходной области, в которой курсор может перемещаться. Затем можно восстановить курсор в исходную область, когда новое ограничение больше не требуется.

## Уничтожение курсора

Можно уничтожить дескриптор курсора и освободить память, используемую курсором, вызвав [функцию DestroyCursor](#). Однако эта функция не влияет на общий курсор. Общий курсор действителен, пока модуль, из которого он был загружен, остается в памяти. Следующие функции получают общий курсор:

- [LoadCursor](#)
- [LoadCursorFromFile](#)
- [LoadImage](#) (если вы используете флаг `LR_SHARED` )
- [CopyImage](#) (если вы используете флаг `LR_COPYRETURNORG`, а *hImage* — общий курсор)

Если вам больше не нужен курсор, созданный с помощью [функции CreateIconIndirect](#) , необходимо уничтожить курсор. Функция [DestroyIcon](#) уничтожает дескриптор курсора и освобождает любую память, используемую курсором. Используйте эту функцию только для курсоров, созданных с помощью [CreateIconIndirect](#).

## Дублирование курсора

Функция [CopyCursor](#) копирует дескриптор курсора. Это позволяет приложению или dll-коду извлекать дескриптор на курсор, принадлежащий другому модулю. Затем, если другой модуль освобожден, модуль, скопированный курсором, по-прежнему может использовать конструктор курсора.

Сведения о добавлении, удалении или замене ресурсов курсора в исполняемых файлах см. в разделе ["Ресурсы"](#).

## Курсор класса Window

При регистрации класса окна с помощью [функции RegisterClass](#) его можно назначить курсор по умолчанию, известный как *курсor* класса. После регистрации класса окна каждое окно этого класса имеет указанный курсор класса.

Чтобы переопределить курсор класса, обработайте [сообщение WM\\_SETCURSOR](#) . Можно также заменить курсор класса с помощью [функции SetClassLong](#). Эта функция изменяет параметры окна по умолчанию для всех окон указанного класса. Дополнительные сведения см. в разделе "[Курсор](#) класса".