

SystemInformation v.1.6.1

Описание настроек



Описание составил *dron3000* на основе материалов разработчика *Daniel Schumann* справочных данных *namut* (<http://gps-club.ru/>) отдельное спасибо за помощь *vovanich* (<http://gps-club.ru/>)

Сайт программы => http://www.ultimatelaunch.de/index_en.html

Скачать программу => http://www.ultimatelaunch.de/download/SystemInformation1.6.1_Menu.zip

Описание автора программы на немецком языке => <http://www.ultimatelaunch.de/systeminformation/inibeschreibung.html>

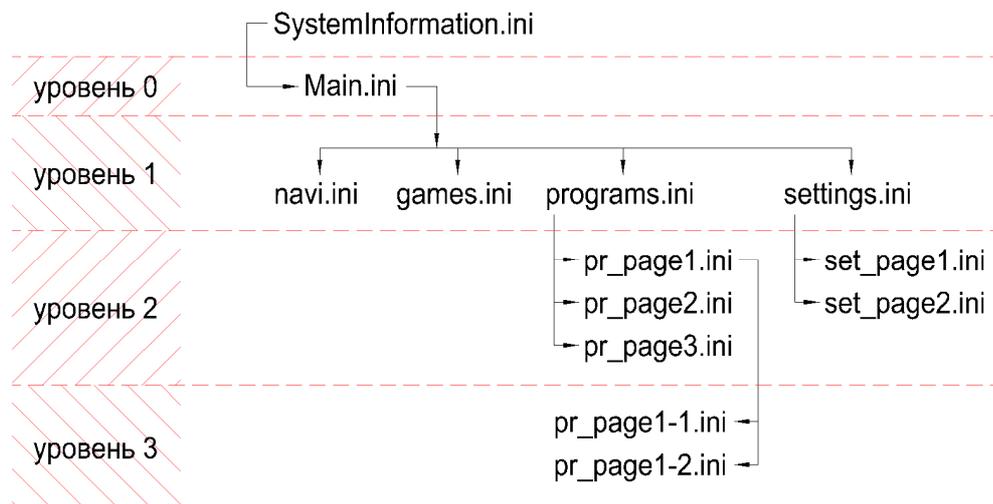
1. Общая информация

В папке скина должны быть следующие файлы:

- *SystemInformation32.exe* – эмулятор меню для тестирования на ББ (для работы программы на навигаторе не нужен);
- *SystemInformation.exe* – файл запуска меню на навигаторе;
- *SystemInformation.ini* – файл настройки, в данном файле указывается путь к основному **.ini** файлу меню (используются абсолютные и относительные пути. Переименование файла не допускается).

Файлы **.ini** должны быть сохранены в формате «Юникод».

Пример структуры меню:



Переход из одного уровня в другой осуществляется путем задания значения параметра **NewIni = [строка]**, см. раздел «Кнопка» элемент **BUTTON**. Переход может быть выполнен из как из высшего уровня в низший, так и наоборот.

Любой **.ini** файл состоит из двух частей:

- Основные настройки окна (размер окна, положение окна относительно других окон, эффекты анимации при вызове/закрытии окна и т.д.), элемент **GLOBALSETTINGS**;
- Описание кнопок и т.д.

РЕКОМЕНДАЦИИ:

- В **.ini** файле «уровня 0» прописать кнопку выхода из меню (параметр **QuitOnExecute = yes**);
- Прописывать для каждого элемента комментарии. Строка, начинающаяся со знака «#» игнорируется;
- По возможности обходится минимальным количеством уровней вложений – *меню => подменю => подподменю => и т.д.* Иначе есть вероятность, что вы просто заблудитесь в лабиринтах собственного меню.
- Для обеспечения гарантированного срабатывания кнопок при нажатии пальцем, задавать минимальный размер кнопки не менее 30x30 (при использовании стилуса размер м.б. меньше);
- Не использовать в качестве фонового рисунка картинки с высокой детализацией, фотографии и т.п., так как это мешает общему восприятию картинки меню. Для фона лучше использовать градиенты, текстуры материалов и т.п.
- Задавать абсолютный путь до исполняемого файла. Для изображения рисунка кнопки или **.ini** файла задавать относительный путь, если изображения кнопок и **.ini** файлы находятся в подпапке меню SystemInformation.
- Параметр *command* задавать в конце описания элемента, иначе есть вероятность, что работать не будет.

2. Общая информация

- Цветовую схему можно посмотреть по ссылке <http://www.uni-magdeburg.de/counter/rgb.txt.shtml>
- Привязка кнопки выполняется по следующему принципу:



- Пример ссылки на исполняемый файл (аналогично задаются пути для изображений кнопок, .ini файлов):
 Ø Абсолютный путь – "`\sdmmc\igo8\iGO.exe`"
 Ø То же, относительный путь задается следующим образом:
 Допустим файл iGO.exe находится по адресу → `\sdmmc\igo8\iGO.exe`
 Меню находится по адресу → `\sdmmc\data\menu_SI\SystemInformation.exe`
 Относительный путь выглядит следующим образом → "`../../iGO.exe`"
`../` → искать в папке в которой расположен файл `SystemInformation.exe` (в папке `menu_SI`);
`../../` → искать в папке расположенной выше папки с файлом `SystemInformation.exe` (в папке `data`);
`../../../../` → искать в папке выше папки с файлом `SystemInformation.exe` на 2 уровня (на диске `sdmmc`) и т.д.
 Запись `../` прописывается необходимый уровень вверх от текущего местоположения файла `SystemInformation.exe`.
- Примеры значений параметров:

Тип	Описание	Примеры
[число]	Может принимать только целые значения:	10 32 -42
[bool]	Значения отличные от 1, Y, YES воспринимаются как «ложь»	No Yes
[строка]	Один или несколько символов	hh':mm':ss Courier New \background.bmp
[цвет]	Значение цвета указывается в шестнадцатеричной системе исчисления (RRGGBB)	FF0000 00FF00 0C43E8
[анимация]	Возможные варианты:	FlyLeft FlyRight FlyUp FlyDown FlyDiagLeftUp FlyDiagLeftDown FlyDiagRightUp FlyDiagRightDown RollDown NoAnimation

3. Основные настройки

GLOBALSETTINGS

Определяет основные настройки окна. Данный элемент должен быть в каждом **.ini** файле меню, параметры в каждом **.ini** файле могут отличаться друг от друга.

- ü **ShowMessage = 0**
? Отображение сообщений (**назначение не известно**)
- ü **BackgroundColor = [цвет]**
Определяет цвет фона
- ü **Width = [число]**
Задаёт ширину окна в пикселях.
- ü **Height = [число]**
Задаёт высоту окна в пикселях.
- ü **X = [число]**
Задаёт расстояние от *левого* края экрана в пикселях. Если значение отрицательное – от *правого* края экрана.
- ü **Y = [число]**
Задаёт расстояние от *верхнего* края экрана в пикселях. Если значение отрицательное – от *нижнего* края экрана.
- ü **BackgroundBitmap = [строка]**
Фоновый рисунок. В параметре указывается абсолютный путь к файлу рисунка.
- ü **BitmapOffsetX = [число]**
Отступ от края экрана по оси X.
Подробнее см. <http://www.ultimatelaunch.de/systeminformation/beschreibung/globalsettings.html>
- ü **BitmapOffsetY = [число]**
Отступ от края экрана по оси Y.
Подробнее см. <http://www.ultimatelaunch.de/systeminformation/beschreibung/globalsettings.html>
- ü **Border = [число]**
Определяет толщину границы вокруг окна в пикселях. Значение должно быть положительным.
- ü **Arrange = [значение]**
Задаёт автоматическое выравнивание. Значение имеет следующие варианты:
Vertical – размещение элементов в один столбец;
Horizontal – размещение элементов в одну строку (не работает);
No – без выравнивания (значение с маленькой буквы *no* программа почему то не понимает).
- ü **Distance = [число]**
Определяет расстояние между отдельными элементами в пикселях, если используется автоматическое выравнивание.
- ü **QuitOnDoubleClick = [Bool]**
Разрешает (*yes*) / запрещает (*no*) выход из меню при двойной клике на свободном поле экрана.
- ü **StartUpAnimation = [анимация]**
Определяет эффект анимации при старте программы.
- ü **ShutDownAnimation = [анимация]**
Определяет эффект анимации при выходе из программы.
- ü **Accelerated = [Bool]**
Разрешает (*yes*) / запрещает (*no*) улучшение анимации.
- ü **AnimationTime = [число]**
Определяет время длительности анимации в миллисекундах.
- ü **TopMost = [Bool]**
Разрешает (*yes*) / запрещает (*no*) режим отображения активного окна поверх всех окон.

4. Текст

TEXT

Вывод на экран произвольного текста.

- **Color = [цвет]**
Определяет цвет текста.
- **FrameColor = [цвет]**
Определяет цвет контура текста.
- **Text = [строка]**
Выводит на экран текст, указанный в параметре.
- **Font = [строка]**
Задаёт тип используемого шрифта. В значении указывается **имя шрифта**. Если данный параметр отсутствует используется шрифт по умолчанию.
- **Size = [число]**
Задаёт размер шрифта в пикселях.
- **Weight = [число]**
Задаёт толщину шрифта. При значениях более **700** – жирный шрифт.
- **CreateFont**
При отсутствии данной строки параметр **Size** не учитывается.
- **X = [число]**
Задаёт расстояние от *левого* края экрана в пикселях. Если значение отрицательное – от *правого* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- **Y = [число]**
Задаёт расстояние от *верхнего* края экрана в пикселях. Если значение отрицательное – от *нижнего* края экрана. Игнорируется в случае автоматического позиционирования элементов.

Пример элемента:

НАВИГАЦИЯ

#текст к кнопке Навигация

TEXT

x = 40

y = 120

Text = НАВИГАЦИЯ

Color = FFFFFFFF

FrameColor = FF4040

Size = 24

Weight = 700

5. Время

TIME

Вывод часов.

ü **Format** = [строка]

Определяет формат вывода часов.

Пример:

Format = hh'ч. 'mm'мин. 'ss'сек.'

на экран выводится в следующем виде:

23ч. 59мин. 03 сек.

Между апострофами (знак ‘) указывается любой текст.

Величины формата времени

Величина	Описание
h	Часы, ноль, для однозначной цифры, не выводится, 12-часовой формат
hh	Часы, 12-часовой формат
H	Часы, ноль, для однозначной цифры, не выводится, 24-часовой формат
HH	Часы, 24-часовой формат
m	Минуты, ноль, для однозначной цифры, не выводится
mm	Минуты
s	Секунды, для однозначной цифры, не выводится
ss	Секунды

ü **Interval** = [число]

Периодичность обновления в миллисекундах, *Interval* = 1000 соответствует 1-ой секунде. По умолчанию периодичность обновления – каждые 60 секунд.

Остальное форматирование аналогично элементу TEXT

Пример элемента:

#часы

TIME

x = 10

y = 10

Color = FF4040

FrameColor = A52A2A

Size = 20

Format = hh'ч. 'mm'мин. 'ss'сек.'

Weight = 500

Interval = 1000

6. Дата

DATE

Вывод даты.

- **Format = [строка]**
Определяет формат вывода даты.
Пример:

Format = d'-dd'-ddd'-dddd

на экран выводится в следующем виде:

7-07-сб-суббота

Между апострофами (знак `'`) указывается любой текст.

Величины формата даты

Величина	Описание
d	Текущее число, ноль, для однозначной цифры, не выводится
dd	Текущее число
ddd	Сокращенное название дня недели
dddd	Полное название дня недели
M	Номер месяца, ноль, для однозначной цифры, не выводится
MM	Номер месяца
MMM	Сокращенное название месяца
MMMM	Полное название месяца
y	Последние две цифры года, ноль, для однозначной цифры, не выводится
yy	Последние две цифры года
yyy	Все цифры года

- **Interval = [число]**
Периодичность обновления в миллисекундах. По умолчанию периодичность обновления каждые 10 минут.

Остальное форматирование аналогично элементу [TEXT](#)

Пример элемента:

#часы

DATE

x = 10

y = 35

Format = d'-dd'-ddd'-dddd

Color = 2F4F4F

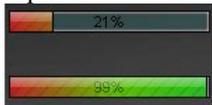
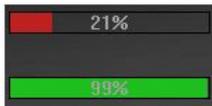
Size = 19

Weight = 700

7. Объем оперативной памяти

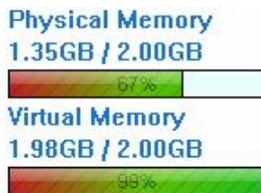
MEMORY

- ü **Color = [цвет]**
Определяет цвет для *TextColor* и *BorderColor*
- ü **TextColor = [цвет]**
Определяет цвет текста.
- ü **FrameColor = [цвет]**
Определяет цвет контура шкалы.
- ü **BackgroundColorLow = [цвет]**
Задаёт цвет шкалы при оставшейся свободной памяти менее 25%.
- ü **BackgroundColorHigh = [цвет]**
Задаёт цвет шкалы при оставшейся свободной памяти более 25%.
- ü **PhysMemString = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение текста «Physical Memory».
- ü **PhysMemBar = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение шкалы физической памяти.
- ü **CoolPhysMemBar = [Bool]**
Определяет внешний вид отображения шкалы (см. рис.)

CoolPhysMemBar = yes	
CoolPhysMemBar = no PhysMemBar = yes	
- ü **PhysMemValue = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение количественного значения занятого/максимального объема физической памяти.
- ü **PhysMemPercent = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение процентов на фоне шкалы виртуальной памяти.
- ü **VirtMemString = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение текста «Physical Memory».
- ü **VirtMemBar = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение шкалы виртуальной памяти.
- ü **CoolVirtMemBar = [Bool]**
Определяет внешний вид отображения шкалы (аналогично **CoolPhysMemBar**)
- ü **VirtMemValue = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение количественного значения занятого/максимального объема виртуальной памяти.
- ü **VirtMemPercent = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение процентов на фоне шкалы виртуальной памяти.
- ü **CenterValue = [Bool]**
Задаёт выравнивание текста названия и значения контролируемого параметра по центру индикаторной шкалы. Если *CenterValue = no* выравнивание выполняется по левому краю индикаторной шкалы.
- ü **Interval = [число]**
Периодичность обновления данных в миллисекундах. По умолчанию обновление выполняется каждые 10 минут.

- ü **X = [число]**
Задает расстояние от *левого* края экрана в пикселях. Если значение отрицательное – от *правого* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Y = [число]**
Задает расстояние от *верхнего* края экрана в пикселях. Если значение отрицательное – от *нижнего* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Width = [число]**
Задает ширину шкалы в пикселях.
- ü **Height = [число]**
Задает высоту шкалы в пикселях.

Пример отображения на экране:



#память

MEMORY

x = -24

y = -50

Width = 144

CoolPhysMemBar = yes

PhysMemBar = yes

PhysMemValue = yes

PhysMemString = yes

CoolVirtMemBar = yes

VirtMemString = yes

VirtMemValue = yes

VirtMemPercent = yes

CenterValue = no

TextColor = 1874CD

BorderColor = 000000

BackgroundColorLow = BF2020

BackgroundColorHigh = 20BF20

Для отображения только физической памяти параметры *CoolVirtMemBar*, *VirtMemString*, *VirtMemValue*, *VirtMemPercent* убрать из описания элемента, либо поставить значение **no**.

Для отображения только виртуальной памяти в параметрах *CoolPhysMemBar*, *PhysMemBar*, *PhysMemValue*, *PhysMemString* поставить значение **no**, если параметры стереть из описания элемента статус физической памяти все равно будет выведен на экран.

8. Объем выбранного диска

DISKMEM

- ü **Color = [цвет]**
Задает цвет для *TextColor* и *BorderColor*.
- ü **TextColor = [цвет]**
Задает цвет текста.
- ü **FrameColor = [цвет]**
Задает цвет контура шкалы.
- ü **BackgroundColorLow = [цвет]**
Задает цвет шкалы при заполнении памяти менее 25%.
- ü **BackgroundColorHigh = [цвет]**
Задает цвет шкалы при заполнении памяти более 25%.
- ü **Folder = [строка]**
Задает имя диска за которым ведется наблюдение (SDMMC, ResidentFlash, SLCFlash и т.д., для тестирования на ББ можно указать *Folder = C:/*).
- ü **ShowPath = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение названия контролируемого диска.
- ü **Bar = [Bool]**
Не работает? <http://www.ultimatelaunch.de/systeminformation/beschreibung/diskmem.html>
- ü **CoolBar = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение шкалы.
- ü **Percent = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение процентов на фоне шкалы.
- ü **Value = [Bool]**
Разрешает (*yes*) / запрещает (*no*) отображение количественного значения *занятого/максимального* объема диска.
- ü **CenterValue = [Bool]**
Задает выравнивание текста названия и значения контролируемого параметра по центру индикаторной шкалы. Если *CenterValue = no* выравнивание выполняется по левому краю индикаторной шкалы.
- ü **Interval = [число]**
Задает периодичность обновления данных в миллисекундах. По умолчанию обновление выполняется каждые 10 минут.
- ü **X = [число]**
Задает расстояние от *левого* края экрана в пикселях. Если значение отрицательное – от *правого* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Y = [число]**
Задает расстояние от *верхнего* края экрана в пикселях. Если значение отрицательное – от *нижнего* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Width = [число]**
Задает ширину шкалы в пикселях.
- ü **Height = [число]**
Задает высоту шкалы в пикселях.

9. Leerraum

? См. <http://www.ultimatelaunch.de/systeminformation/beschreibung/space.html>

SPACE

ü Space = [число]
?

10. Батарея

BATTERY

- ü **ColorBattery = [цвет]**
Задает цвет фона батареи.
- ü **ColorCritical = [цвет]**
Задает цвет минимального заряда батареи – 0...33%.
- ü **ColorLow = [цвет]**
Задает цвет среднего заряда батареи – 34...66%.
- ü **ColorHigh = [цвет]**
Задает цвет максимального заряда батареи – 67...100%.
- ü **CoolBattery = [Bool]**
Определяет внешний вид отображения батареи (см. рис.)

CoolBattery = yes



CoolBattery = no



- ü **ColorArrow = [цвет]**
Определяет цвет молнии на батарейке
- ü **Segments = [число]**
Количество сегментов батареи.
- ü **Test = [Bool]**
Разрешает (*yes*) / запрещает (*no*) режим тестирования. В режиме тестирования поочередно показываются все стадии заряда батареи.
- ü **Interval = [число]**
Задает период обновления данных в миллисекундах. По умолчанию обновление выполняется каждые 10 минут.
- ü **X = [число]**
Задает расстояние от *левого* края экрана в пикселях. Если значение отрицательное – от *правого* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Y = [число]**
Задает расстояние от *верхнего* края экрана в пикселях. Если значение отрицательное – от *нижнего* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Width = [число]**
Задает ширину шкалы в пикселях.
- ü **Height = [число]**
Задает высоту шкалы в пикселях.

11. Кнопка

BUTTON

Создает невидимую активную область, при клике на которой, выполняется команда.

- ü **Command** = [строка]
Вызов программы. В значении указывается абсолютный путь до исполняемого файла.
Пример:
Command = *lsdmmc\MENU_S\Utils\PowerOff.exe*
- ü **NewIni** = [строка]
Обращение к другой странице меню.
Пример:
NewIni = *..Main\Games.ini*
- ü **ShutDownAnimation** = [анимация]
Задаёт эффект анимации.
- ü **QuitOnExecute** = [Bool]
Разрешает (*yes*) / запрещает (*no*) выход из программы при клике на кнопке.
Если в описании кнопки присутствует параметр *Command* = [строка], меню закроется после запуска указанной в *Command* программы.
- ü **X** = [число]
Задаёт расстояние от *левого* края экрана в пикселях. Если значение отрицательное – от *правого* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Y** = [число]
Задаёт расстояние от *верхнего* края экрана в пикселях. Если значение отрицательное – от *нижнего* края экрана. Игнорируется в случае автоматического позиционирования элементов.
- ü **Width** = [число]
Задаёт ширину активной зоны.
- ü **Height** = [число]
Задаёт высоту активной зоны.

Пример элемента:

```
#вызов Taskbar
```

```
BUTTON
```

```
x = 0
```

```
y = 0
```

```
Height = 12
```

```
Width = 320
```

```
Command = "\SDMMC\ShowTaskbar.exe"
```

12. Кнопка IconXP

ICONXPBUTTON

Для отображения внешнего вида кнопки используются файлы **.ico**.

- ü **SizeNormal = [число]**
Задаёт размер кнопки в состоянии покоя. Значение соответствует размеру иконки из файла **.ico**.
- ü **SizePushed = [число]**
Задаёт размер кнопки при нажатии. Значение соответствует размеру иконки из файла **.ico**.
- ü **IconNormal = [строка]**
Задаёт путь к файлу с изображением кнопки в состоянии покоя.
- ü **IconPushed = [строка]**
Задаёт путь до файла с изображением кнопки в нажатом состоянии.
- ü **Gray = [Bool]**
После нажатия, кнопка становится серой.
- ü Влияние параметров **ScaleAlpha, ScaleRed, ScaleGreen, ScaleBlue, AddRed, AddGreen, AddBlue** на внешний вид кнопки (см. таблицу)

 Параметр = [число]
 Указывается положительное число.

Изображение кнопки без применения фильтров =>



Параметр/значение	20%	50%	80%	100%
ScaleAlpha				

Параметр/значение	0%	50%	100%	200%
ScaleRed				
ScaleGreen				
ScaleBlue				

Параметр/значение	0%	50%	100%	200%
AddRed				
AddGreen				
AddBlue				

Остальные параметры аналогично элементу [BUTTON](#), параметры X, Y обязательны.

Пример элемента:

#выход из меню

ICONXPBUTTON

x = -20

y = -10

QuitOnExecute = yes

SizeNormal = 48

SizePushed = 64

IconNormal = ../icons/off-power.ico

IconPushed = ../icons/off-power-on.ico

13. Кнопка Bitmap

BITMAPBUTTON

Для отображения внешнего вида кнопки используются файлы с растровыми изображениями **.bmp**.

- **BitmapNormal** = [строка]
Задаёт путь к файлу с изображением кнопки в состоянии покоя.
- **BitmapPushed** = [строка]
Задаёт путь к файлу с изображением кнопки в нажатом состоянии.
- **Offset** = [число]
Задаёт смещение изображения кнопки в пикселях при нажатии на нее.

Остальные параметры аналогично элементу BUTTON, параметры **X**, **Y**, **Width**, **Height** обязательны.

Пример элемента:

```
#кн настройка
BITMAPBUTTON
x = -50
y = 50
Height = 48
Width = 48
BitmapNormal = ../icons/setup_48.bmp
BitmapPushed = ../icons/setup_48.bmp
Offset = 1
```

14. Кнопка Windows

WINDOWSBUTTON

Кнопка Windows

Кнопка Windows

Упрощенный вид кнопки, рисунок для кнопки не требуется.

- ü **ColorNormal** = [цвет]
Задаёт цвет кнопки в состоянии покоя.
- ü **ColorPushed** = [цвет]
Задаёт цвет кнопки в нажатом состоянии.
- ü **Text** = [строка]
Вывод текста на фоне кнопки.

Остальные параметры аналогично элементу BUTTON, параметры **X**, **Y**, **Width**, **Height** обязательны.

Пример элемента:

```
#кнопка info
WINDOWSBUTTON
x = 135
y = 90
Width = 60
Height = 20
Text = Кнопка Windows
ColorNormal = FF4040
ColorPushed = 8B2323
NewIni = ../Info/info.ini
```