

{{>TOC}}

Формат *.ECD

Формат *.ecd (*Exarion Cascade Descriptor*) по сути является текстовым файлом в кодировке UTF-8. (Это важно, использование любой другой кодировки может привести к ошибкам чтения и как следствие к ошибкам в логике приложения или даже стать причиной исключений). Для работы с файлом рекомендуется использовать редактор [Notepad++](#), для версии 6.3+ была сделана подсветка синтаксиса формата *.ecd. Для установки скачайте файл [ecd.xml](#) далее в главном меню выберите "Синтаксис" -> "Задать свой язык..." в появившемся окне нажмите кнопку "Импорт" и выберите скаченный файл ecd.xml. Теперь можно закрывать окно, в меню "Синтаксис" должно было появиться в самом низу пункт "ECD", при выборе которого будет применена для текущего файла подсветка синтаксиса.

Описание файлов *.ecd

| {{child_pages(Формат_ECD)}}

Основной синтаксис

В простейшем случае файл состоит из последовательности пар **ключ-значение**, разделяемых двоеточием, таким образом значением является любые символы начиная с двоеточия и до следующей управляющей конструкции. Перед ключем ставится символ бакса '\$'. При чтении все пустые символы игнорируются (т.е. пробелы, табуляция и т.д.), следовательно их можно спокойно использовать для форматирования текста для его более наглядного визуального представления. Синтаксис не чувствителен к регистру, а возможные значения ключей зависят от программного обработчика файла (т.е. допустимые ключи могут меняться в разных файлах или даже в его частях). Каждому ключу можно присвоить только одно значение, возможны следующие типы простых значений:

- Логический тип - это значение, допускающее всего два варианта "да" или "нет". Для большей наглядности текста могут использоваться несколько вариантов обозначения: true/false, yes/no, on/off.
- Целочисленное число - может содержать в начале знак "+" или "-" а так же начинаться с приставки "#" в этом случае число является шестнадцатеричным. По умолчанию считается что число положительное в десятичной системе.
- Не целочисленное число - обязательно должно содержать дробную часть, отделяемую точкой (использование запятой вместо точки является ошибкой), так же как и целочисленное число может содержать в начале знак "+" или "-".
- Строка (текст) - последовательность символов составляющая строку должна быть заключена в двойные кавычки. Внутри кавычек допускается использование любых символов, кроме самих кавычек и квадратных скобок ('[', ']'), в том числе и перенос строк. Строка может начинаться с локали "ru." \ "en." определяющей принадлежность строки к тому или иному языку. В случае если локали нет то строка является универсальной для любого языка.
- Массив (набор значений) - в некоторых случаях ключи могут требовать последовательной значений (например координата состоит из значения X и Y), в этом случае последовательной заключается в фигурные скобки, а значения разделяются запятыми. Массив обязательно должен состоять как минимум из двух значений.

Порой полезно сделать какие-то пометки для себя или для других, для этого используются комментарии. Комментарий начинается с символа "~" и заканчивается с началом новой строки.

Пример:

```
1 $btnEnable: true           ~ Логический тип, определяющий активность кнопки
2 $btnRect:{300,220,200,40}  ~ Массив задающий расположение (x,y,ширина,высота)
3 $btnScale: 0.5             ~ Масштаб кнопки
4 $btnSoundId: #001A         ~ Id используемого звук (#001A = 26)
5 $btnImage:"pic_menu_button" ~ Название файла с текстурой
6 $btnLabel:RU."Новая Игра"  ~ Текст кнопки для русского языка
7 $btnLabel:EN."New Game"    ~ Текст кнопки для английского языка
```

Файлы *.ecd читаются последовательно, поэтому порядок расположения элементов может быть важен, например в этом случае:

```
1 $btnLabel:RU."Новая Игра"  ~ Текст кнопки для русского языка
2 $btnLabel:EN."New Game"    ~ Текст кнопки для английского языка
3 $btnLabel:"Unknown Button" ~ Текст кнопки для остальных языков
```

Значение текста кнопки всегда будет "Unknown Button", поскольку последнее описание переопределяет сделанное ранее. Что бы это исправить надо поместить последнюю строку в начало. В некоторых случаях, таких как описание заставок, файлы являются пакетными постепенно. Более подробно о ключах и форматировании смотрите в описании конкретных файлов.

Экранирование символов

Для использования особых символов в строках служит спец символ '\', что преобразует следующий за ним символ по следующему правилам:

Исходная комбинация	\'	\<	\>	\-	*	\p	\a	\n	
Преобразуется парсером	"	[]	~	\	%	@	новая строка	

Расширения синтаксиса

Ниже приводятся расширения синтаксиса, требуемые для описание некоторых данных. Следует помнить что логика работы расширений целиком зависит от конкретного файла, тут лишь описывается общий синтаксис.

Секции

Выше был рассмотрен простейший вариант, с линейной структурой. В некоторых случаях файл может быть разделен на секции, каждая секция не зависима и может рассматривать как отдельный файл. Началом новой секции служит ее название в квадратных скобках. После объявления секция будет продолжаться либо до объявления следующей секции либо до конца файла. Пример:

- 1 \$btnLabel:"Next" ~ Текст кнопки для меню игры
- 2 [HighScore]
- 3 \$btnLabel:"Return" ~ Текст кнопки для страницы счета в меню игры

Блоки

todo

Макросы

todo

Функции

todo

Действия

todo