

# Генератор символики Aztec

ОПИСАНИЕ

ТИПЫ

*ag\_matrix*

*ag\_settings*

ФУНКЦИИ

*ag\_generate*

*ag\_release\_matrix*

КОДЫ ОШИБОК

ПРИМЕР ИСПОЛЬЗОВАНИЯ

## **Описание**

Библиотека предназначена для создания штриховых кодов «Aztec Code». Она поддерживает C- и Delphi-интерфейсы и распространяется под т.н. «модифицированной» BSD-лицензией (текст лицензии приведён в файле license.txt).

# Типы

## ag\_matrix

Матрица точек, где 0 – светлая точка, 1 – тёмная, описывающая штриховой код.

```
typedef struct ag_matrix_tag
{
    unsigned char *data;
    size_t width;
    size_t height;
}
ag_matrix;
```

Поля:

<i>data</i>	Указатель на массив точек. Точки располагаются построчно, строки не выровнены
<i>width</i>	Количество столбцов
<i>height</i>	Количество строк

## ag\_settings

Настройки генерации матрицы точек.

```
typedef struct ag_settings_tag
{
    unsigned long mask;
    unsigned char symbol_type;
    unsigned char symbol_format;
    unsigned char redundancy_for_error_correction;
}
ag_settings;
```

Поля:

<i>mask</i>	Маска задействованных полей. Может быть равна одному из значений AG_SF_... или быть их комбинацией через операцию «побитовое ИЛИ»
<i>symbol_type</i>	Тип символа. Принимает одно из значений AG_..._SYMBOL (по умолчанию – AG_NORMAL_SYMBOL)
<i>symbol_format</i>	Формат символа. Принимает одно из значений AG_..._FORMAT (по умолчанию – AG_ANYONE_FORMAT)
<i>redundancy_for_error_correction</i>	Минимальный процент места в символе, отведённого под слова коррекции ошибок. Принимает значение от 0 до 99 (по умолчанию – 23)

## Функции

### ag\_generate

Генерация матрицы точек.

```
AG_API(int) ag_generate(ag_matrix **matrix, const void *data,  
                        size_t data_size, const ag_settings *settings);
```

Параметры:

<i>matrix</i>	Указатель, по которому записывается указатель на сгенерированную матрицу точек
<i>data</i>	Указатель на кодируемые данные
<i>data_size</i>	Размер кодируемых данных в байтах
<i>settings</i>	Указатель на структуру с настройками генерации. Если он равен NULL, то используются настройки по умолчанию

Возвращаемое значение:

См. Коды ошибок.

### ag\_release\_matrix

Освобождение памяти, выделенной под матрицу точек.

```
AG_API(int) ag_release_matrix(ag_matrix *matrix);
```

Параметры:

<i>matrix</i>	Указатель на освобождаемую матрицу точек
---------------	--

Возвращаемое значение:

См. Коды ошибок.

**Коды ошибок**

Название	Значение	Описание
AG_SUCCESS	0	Ошибок нет
AG_INVALID_PARAMETER	1	Неверный входной параметр
AG_OUT_OF_MEMORY	2	Нехватка памяти
AG_UNHANDLED_EXCEPTION	3	Необработанное исключение
AG_NO_INPUT_DATA	4	Отсутствуют входные данные
AG_TOO_MUCH_INPUT_DATA	5	Слишком много входных данных

## Пример использования

Исходный код:

```
#include <stddef>
#include <stdio>
#include "aztecggen.h"

int main()
{
    enum {success, error};

    const char data[] = "Hello world :)";
    enum {symbol_format = AG_FULL_FORMAT};

    // Barcode creation
    ag_settings settings;
    settings.mask = AG_SF_SYMBOL_FORMAT;
    settings.symbol_format = symbol_format;

    ag_matrix *barcode;
    const int gen_result = ag_generate(
        &barcode, data, sizeof(data) - 1, &settings);
    if (gen_result != AG_SUCCESS)
    {
        printf("Can't create a barcode (error code: %i)\n",
            gen_result);
        return error;
    }

    // Barcode printout
    printf("Data:\n");
    printf("\t%s\n", data);

    printf("Barcode:\n");
    for (size_t y = 0; y < barcode->height; ++y)
    {
        printf("\t");
        for (size_t x = 0; x < barcode->width; ++x)
        {
            const unsigned char t =
                barcode->data[y * barcode->width + x];
            printf("%c", (t == 0) ? ' ': '#');
        }
        printf("\n");
    }
    printf("\n");

    // Barcode destruction
    ag_release_matrix(barcode);

    return success;
}
```

Вывод:

Data:

Hello world :)

Barcode:

```
# # ##          # # ### #
### # ##### ##### #
# # ## ###    ## # # #
### # ##### ##  ### #
#          ##    ##  ###
# # #####          #
### #          ##  #
## # # ##### ##### #
# ##### #          # # #
# # # # ##### # #
# # # # # # # # # # #
##### # # # # # # #
### # ##### # #####
# # #          # #
# ### ##### #####
# #####          # ##
#####          #
### ##          # #
### ### ##### # ## #
# # ## ##    #    ## # #
##### #    # #####
##### # # # # # # #
```