# utf8_txt.c File Reference

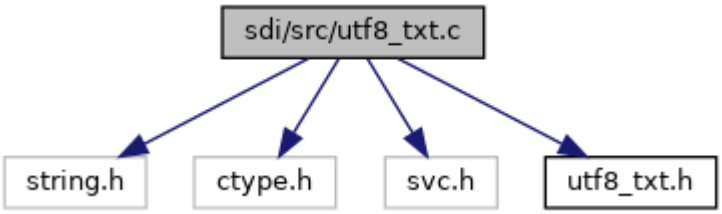
```
#include <string.h>

#include <ctype.h>

#include "svc.h"

#include "utf8_txt.h"
```

Include dependency graph for utf8_txt.c:




## Data Structures

| | |
|---|---|
| struct | LangMapTable |

## Functions

| | |
|---|---|
| enum Language | getInternalLanguageId (unsigned char externalLanguageId) |
| enum Language | getExternalLanguageId (unsigned char internalLanguageId) |
| void | getLanguageIsoCode (enum Language languageId, unsigned char *languageIsoCode) |
| enum Language | getLanguage (unsigned char *ISO_639) |

## Variables

| | |
|---|---|
| const struct LangMapTable | emvLangMapTable [] |
| const struct LangMapTable | langMapTable [] |

## Data Structure Documentation

## ? LangMapTable

struct LangMapTable

Automatically generated file. DO NOT MODIFY

**Data Fields**

| | |
|---|---|
| unsigned char | ISO_639[3] |
| enum Language | lang |

## Function Documentation

## ? getExternalLanguageId()

enum Language getExternalLanguageId ( unsigned char *internalLanguageId* )

Get the external language corresponding to the ISO 639-1 value

Parameters

[in] internalLanguageId ISO 639-1 internal language id (e.g. de, en, fr)

Returns

enum of the external language

## ? getInternalLanguageId()

enum Language getInternalLanguageId ( unsigned char *externalLanguageId* )

Get the internal language corresponding to the ISO 639-1 value

Parameters

[in] externalLanguageId ISO 639-1 external language Id (e.g. de, en, fr)

Returns

enum of the internal language

## ? getLanguage()

enum Language getLanguage ( unsigned char * *ISO_639* )

Get the language corresponding to the ISO 639-1 value

Parameters

[in] ISO_639 ISO 639-1 value (e.g. de, en, fr)

Returns

enum of the language

## ? getLanguageIsoCode()

void getLanguageIsoCode ( enum Language *languageId*,
unsigned char * *languageIsoCode*
)

Get the ISO 639-1 value corresponding to the language

Parameters

[in] languageId Language Id
[out] languageIsoCode ISO 639-1 value

Returns

ISO 639-1 value (two bytes, zero terminated)

## Variable Documentation

## ? emvLangMapTable

const struct LangMapTable emvLangMapTable[]

**Initial value:**

```
={       {(enum Language)0x01, "en"},   {(enum Language)0x02, "de"},   {(
enum Language)0x03, "fr"},   {(enum Language)0x04, "es"},   {(enum Language
)0x05, "it"},   {(enum Language)0x06, "cs"},   {(enum Language
)0x07, "sk"},   {(enum Language)0x08, "sv"},   {(enum Language
)0x09, "pl"},   {(enum Language)0x0A, "el"},   {(enum Language
)0x0B, "tr"},   {(enum Language)0x0C, "da"},   {(enum Language
)0x0D, "nl"},   {(enum Language)0x0E, "no"},   {(enum Language
)0x0F, "pt"},   {(enum Language)0x10, "at"},   {(enum Language
)0x11, "et"},   {(enum Language)0x12, "fi"},   {(enum Language
)0x13, "lv"},   {(enum Language)0x14, "lt"},   {(enum Language
)0x15, "ru"},   {(enum Language)0x16, "bg"},   {(enum Language
)0x17, "hr"},   {(enum Language)0x18, "hu"},   {(enum Language
)0x19, "mo"},   {(enum Language)0x1A, "ro"},   {(enum Language
)0x1B, "sr"},   {(enum Language)0x1C, "sl"},   {(enum Language
)0x1D, "he"}  }
```

## ? langMapTable

const struct LangMapTable langMapTable[]

**Initial value:**

```
={  {LANG_ENGLISH,    "en"},  {LANG_GERMAN,     "de"},  {LANG_FRENCH,
"fr"},  {LANG_SPANISH,    "es"},  {LANG_ITALIAN,    "it"
},  {LANG_SWEDISH,    "sv"},  {LANG_TURKISH,    "tr"},  {LANG_DANSK,
"da"},  {LANG_DUTCH,      "nl"},  {LANG_PORTUGUESE, "pt"
},  {LANG_HEBREW,     "he"}}
```