#pragma once

#include <twml/defines.h>

#include <stddef.h>

#include <stdint.h>

#ifdef \_\_cplusplus

namespace twml {

template<typename T> struct Type;

template<> struct Type<float>

{

enum {

type = TWML\_TYPE\_FLOAT,

};

};

template<> struct Type<std::string>

{

enum {

type = TWML\_TYPE\_STRING,

};

};

template<> struct Type<double>

{

enum {

type = TWML\_TYPE\_DOUBLE,

};

};

template<> struct Type<int64\_t>

{

enum {

type = TWML\_TYPE\_INT64,

};

};

template<> struct Type<int32\_t>

{

enum {

type = TWML\_TYPE\_INT32,

};

};

template<> struct Type<int8\_t>

{

enum {

type = TWML\_TYPE\_INT8,

};

};

template<> struct Type<uint8\_t>

{

enum {

type = TWML\_TYPE\_UINT8,

};

};

template<> struct Type<bool>

{

enum {

type = TWML\_TYPE\_BOOL,

};

};

}

#endif