#ifndef TWML\_LIBTWML\_INCLUDE\_TWML\_COMMON\_H\_

#define TWML\_LIBTWML\_INCLUDE\_TWML\_COMMON\_H\_

#define USE\_ABSEIL\_HASH 1

#if defined(USE\_ABSEIL\_HASH)

#include "absl/container/flat\_hash\_map.h"

#include "absl/container/flat\_hash\_set.h"

#elif defined(USE\_DENSE\_HASH)

#include <sparsehash/dense\_hash\_map>

#include <sparsehash/dense\_hash\_set>

#else

#include <unordered\_map>

#include <unordered\_set>

#endif // USE\_ABSEIL\_HASH

namespace twml {

#if defined(USE\_ABSEIL\_HASH)

template<typename KeyType, typename ValueType>

using Map = absl::flat\_hash\_map<KeyType, ValueType>;

template<typename KeyType>

using Set = absl::flat\_hash\_set<KeyType>;

#elif defined(USE\_DENSE\_HASH)

// Do not use this unless an proper empty key can be found.

template<typename KeyType, typename ValueType>

using Map = google::dense\_hash\_map<KeyType, ValueType>;

template<typename KeyType>

using Set = google::dense\_hash\_set<KeyType>;

#else

template<typename KeyType, typename ValueType>

using Map = std::unordered\_map<KeyType, ValueType>;

template<typename KeyType>

using Set = std::unordered\_set<KeyType>;

#endif // USE\_DENSE\_HASH

} // namespace twml

#endif // TWML\_LIBTWML\_INCLUDE\_TWML\_COMMON\_H\_