

How to determine tag status in the FIX message using FixDictionary API in FIX Antenna C++

The FIX Antenna C++ provides the public API to the FIX dictionaries loaded by the engine that can be used to determine tag status in the FIX message:

```
#include <B2BITS_Dictionary.h>
#include <B2BITS_Protocol.h>
#include <B2BITS_Message.h>
#include <B2BITS_Item.h>
#include <B2BITS_MessageItem.h>
#include <B2BITS_MessageItemContainer.h>
#include <B2BITS_Block.h>
#include <B2BITS_BlockRef.h>
#include <B2BITS_FieldRef.h>
#include <B2BITS_RepeatingGroup.h>

#include <B2BITS_FIXFields.h>

enum TagStatus {TAG_NOT_FOUND, LEADING_TAG, START_TAG, TAG_IN_GROUP };

TagStatus getTagStatusImpl(FixDictionary2::ProtocolT::CPtr protocol, const FixDictionary2::
MessageItemContainer* items, const int tag)
{
    FixDictionary2::MessageItemT::CRefArray childs;
    const std::size_t ccount = items->getChilds(&childs);
    //leading tag is included in both repeating group and field, so test repeating groups first
    for(size_t c = 0; c < ccount; c++)
    {
        FixDictionary2::MessageItemT::CRef& item = childs[c];
        if(FixDictionary2::Item::ITEM_REPEATINGGROUP == item->itemName()){
            if(item->toRepeatingGroup()->getLeadingTag() == tag) {
                return LEADING_TAG;
            } else if(item->toRepeatingGroup()->getStartTag() == tag) {
                return START_TAG;
            } else {
                TagStatus st = getTagStatusImpl(protocol, item->toRepeatingGroup(), tag);
                if(TAG_NOT_FOUND != st) {
                    return st;
                }
            }
        }
    }
    for(size_t c = 0; c < ccount; c++)
    {
        FixDictionary2::MessageItemT::CRef& item = childs[c];
        if(FixDictionary2::Item::ITEM_BLOCKREF == item->itemName()){
            FixDictionary2::BlockT::CPtr block = protocol->getBlock( item->toBlockRef()->getId() );
            if(block) {
                TagStatus st = getTagStatusImpl(protocol, block, tag);
                if(TAG_NOT_FOUND != st) {
                    return st;
                }
            }
        }
    }
    for(size_t c = 0; c < ccount; c++)
    {
        FixDictionary2::MessageItemT::CRef& item = childs[c];
        if(FixDictionary2::Item::ITEM_FIELDREF == item->itemName()){
            if(item->toFieldRef()->getTag() == tag) {
                return TAG_IN_GROUP;
            }
        }
    }
    return TAG_NOT_FOUND;
}
```

```
TagStatus getTagStatus(const std::string& protocolName, const std::string& messageType, const int tag)
{
    FixDictionary2::ProtocolT::CPtr protocol = Engine::FixEngine::singleton()->getDictionary()->getProtocol
(protocolName);
    if(!protocol) {
        return TAG_NOT_FOUND;
    }
    FixDictionary2::MessageT::CPtr message = protocol->getMessage(messageType);
    if(!message) {
        return TAG_NOT_FOUND;
    }
    return getTagStatusImpl(protocol, message, tag);
}

void example()
{
    getTagStatus("FIX44", "D", FIXFields::Account);
}
```