How to work with Repeating Groups

Preface

This page contains code snippets with solutions to some typical tasks for repeating groups.

Retrieving Repeating Group Instance

Use *'getGroup'* method of corresponding parent FIXMessage or nested repeating group.

Example.cpp

```
#include <B2BITS_FIXMessage.h>
#include <B2BITS_FIXGroup.h>
#include <B2BITS_FIXFields.h>
//...

FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);
```

Please note, that memory of FIXGroup instance is managed by it's parent FIXMessage, you should not delete it by yourself.

Getting Repeating Group size (number of entries)

Either get FIXGroup object and get it's size:

Example.cpp

```
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);
int entryCount = ( pGroup == NULL ? 0 : pGroup->size() );
```

or get integer value of corresponding leading tag:

Example.cpp

```
int entryCount = ( pMessage->isEmpty(FIXFields::NoQuoteSets) ? 0 : pMessage->getAsInt(FIXFields::NoQuoteSets) );
```

Iterating over Repeating Group entries and fields

Example 1

Get access to FIXGroup instance, determine it's size,
and then access group fields same way with FIXMessage fields, except you need to specify entry index:

MyApp.cpp

```
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

if ( NULL != pGroup ) {
        for ( int i = 0; i < pGroup->size(); ++i ) {
          std::cout<< pGroup->getAsString(FIXFields::QuoteSetID, i).toStdString() <<"\n";
        }
}
```

Please note that entry indexing starts from zero.

Example 2

Get access to FIXGroup instance, determine it's size,
and then iterate group entries one by one; access entry content same way with FIXMessage fields.

MyApp.cpp

```
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

if ( NULL != pGroup ) {
        for ( int i = 0; i < pGroup->size(); ++i ) {
                TagValue* pEntry = pGroup->getEntry(i);
                std::cout<< pEntry->getAsString(FIXFields::QuoteSetID).toStdString() <<"\n";
        }
}
```

Example 3 (since FIX Antenna 2.19)

Use built-in iterators:

MyApp.cpp

```
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

if ( NULL != pGroup ) {
        FIXGroup::forward_iterator it = pGroup->begin(), end = pGroup->end();
    for (; it != end; ++it ) {
          std::cout<< it->getAsString(FIXFields::QuoteSetID).toStdString() <<"\n";
        }
}
```

or, if your compiler supports C++11:

MyApp.cpp

```
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

if ( NULL != pGroup ) {
    for ( auto it : *pGroup ) {
          std::cout<< it->getAsString(FIXFields::QuoteSetID).toStdString() <<"\n";
        }
}
```

Iterating over Nested Repeating Groups entries and fields

Nested groups are accessed same way as top-level groups.

Example 1

MyApp.cpp

```
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

if ( NULL != pGroup ) {
        for ( int i = 0; i < pGroup->size(); ++i ) {
                std::cout<< pGroup->getAsString(FIXFields::QuoteSetID, i).toStdString() <<"\n";

                FIXGroup* pNestedGroup = pGroup->getGroup(FIXFields::NoQuoteEntries, i);

                if( NULL != pNestedGroup ) {
                        for ( int j = 0; j < pNestedGroup ->size(); ++j ) {
                                std::cout <<"\t" << pNestedGroup ->getAsString(FIXFields::QuoteEntryID, j).
toStdString() <<"\n";
                        }
                }
        }
}
```

Example 2 (since FIX Antenna 2.19)

MyApp.cpp

```
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

if ( NULL != pGroup ) {
        FIXGroup::forward_iterator it = pGroup->begin(), end = pGroup->end();
    for (; it != end; ++it ) {
                std::cout<< it->getAsString(FIXFields::QuoteSetID).toStdString() <<"\n";

                FIXGroup* pNestedGroup = it->getGroup(FIXFields::NoQuoteEntries);
                if ( NULL != pNestedGroup ) {
                        FIXGroup::forward_iterator nit = pNestedGroup ->begin(), nend = pNestedGroup ->end();
                        for (; nit != nend; ++it ) {
                                std::cout <<"\t" << pNestedGroup ->getAsString(FIXFields::QuoteEntryID, j).
toStdString() <<"\n";
                        }
                }
        }
}
```

or, if your compiler supports C++11:

MyApp.cpp

```
FIXGroup* pGroup, pNestedGroup;

if( pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);) {
    for( auto it : *pGroup ) {
                std::cout<< it->getAsString(FIXFields::QuoteSetID).toStdString() <<"\n";

                if ( pNestedGroup = it->getGroup(FIXFields::NoQuoteEntries) ) {
                        for( auto nit : *pNestedGroup ) {
                                std::cout <<"\t" << nit->getAsString(FIXFields::QuoteEntryID).toStdString() <<"
\n";
                        }
                }
    }
}
```

Resizing Repeating Group

Set corresponding leading tag to required size:

MyApp.cpp

```
pMessage->setAsInt(FIXFields::NoQuoteSets, myNewSize);
```

If new size is greater than previous then new entries with empty contents will be added;
If new size is less than previous, then exceeding entries at the end of group will be discarded.
Setting size to zero efficiently deletes group from message.

Adding entries to Repeating Group without knowing their quantity ahead of time

Example 1

Increase current size by 1 manually, fill last entry with values:

MyApp.cpp

```
int noQuoteSets = pMessage->getAsInt(FIXField::NoQuoteSets);
++noQuoteSets;
pMessage->set(FIXField::NoQuoteSets, noQuoteSets);

FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);
TagValue* pEntry = pGroup ->getEntry(noQuoteSets-1);
pEntry->set(FIXFields::QuoteSetID, "abcd");
```

Example 2

Reserve some space in group, fill entries, shrink group to actual size.

MyApp.cpp

```
pMessage->setAsInt(FIXFields::NoQuoteSets, 200); //reserve
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

//iterate over Repeating Group entries, filling values
...

pMessage->set(FIXField::NoQuoteSets, actualNoQuoteSets ); //shrink
```

Example 3 (since FIX Antenna 2.19)

Use helpers from B2BITS_FIXMsgHelper.h:

MyApp.cpp

```
#include <B2BITS_FIXMsgHelper.h>
//...

TagValue* entry = FIXMsgHelper::addEntryToNestedGroup(pMessage, FIXField::NoQuoteSets);
entry->set(FIXField::QuoteSetID, "abcd" );
```

Populating FIX message with nested Repeating Groups

Populating nested repeating groups is mostly same as repeating groups in message.

Example 1

MyApp.cpp

```
// create one entry of NoQuoteSets in FIX message
pMessage->set(FIXField::NoQuoteSets, 1);
FIXGroup* pGroup = pMessage->getGroup(FIXFields::NoQuoteSets);

// create 5 entries of nested group NoQuoteEntries at entry [0] of NoQuoteSets
pGroup->set( FIXField::NoQuoteEntries, 5, 0 );

// get access to NoQuoteEntries nested group
FIXGroup* pNestedGroup = pGroup->getGroup( FIXField::NoQuoteEntries, 0 );

// Fill QuoteEntryID tag in entry 0 of NoQuoteEntries nested group
pNestedGroup ->set( FIXField::QuoteEntryID, "SubEntry_0", 0 );
```

Example 2 (since FIX Antenna 2.19)

Use helpers from B2BITS_FIXMsgHelper.h:

MyApp.cpp

```
#include <B2BITS_FIXMsgHelper.h>
//...

FIXMsgHelper::addEntryToNestedGroup(pMessage, FIXField::NoQuoteSets);
FIXGroup& group = pMessage->getAsGroup(FIXField::NoQuoteSets);

// at entry 0 of NoQuoteSets add entry to NoQuoteEntries nested group
TagValue* embeddedEntry = FIXMsgHelper::addEntryToNestedGroup(group, FIXField::NoQuoteEntries, 0);

// Fill QuoteEntryID tag in new entry of NoQuoteEntries nested group
embeddedEntry->set( FIXField::QuoteEntryID, "SubEntry1");
```

Removing arbitrary entry from Repeating Group

Currently the following approach is recommended to remove entry under index '**i**' from group with total length '**N**':

1. Clear value for every tag in target entry ({$X_{\mathbf{i}}$,$Y_{\mathbf{i}}$,$Z_{\mathbf{i}}$}).
2. Copy new value from last entry: ({$X_{\mathbf{N-1}}$,$Y_{\mathbf{N-1}}$,$Z_{\mathbf{N-1}}$} -> {$X_{\mathbf{i}}$,$Y_{\mathbf{i}}$,$Z_{\mathbf{i}}$})
3. Resize group to '**N-1**', so that last entry ({$X_{\mathbf{N-1}}$,$Y_{\mathbf{N-1}}$,$Z_{\mathbf{N-1}}$}) will be deleted from message.