

```
/*
** RELEASE STATEMENT(s):
**
**             UNLIMITED RIGHTS
** The Government has the right to use, modify, reproduce, release, perform,
** display, or disclose this application programmable interface in whole or in
** part, in any manner and for any purpose whatsoever, and to have or
** authorize others to do so.
**
** Distribution Statement A - Approved for public release; distribution is
** unlimited (27 August 2015).
*/

/*
** JTNC Standard:
** Software Communications Architecture
** Appendix E: Model Driven Support Technologies
** Version: 4.1, 20 August 2015
*/

/* File: CFSequenceTemplates.h */
#ifndef __CFSEQUENCETEMPLATES_DEFINED
#define __CFSEQUENCETEMPLATES_DEFINED
#include "CFPrimitiveTypes.h"

namespace CF
{
    template <class T>
    class Sequence
    {
    public:
        Sequence();
        Sequence(CF::ULongType max);
        Sequence(
            CF::ULongType max,
            CF::ULongType length,
            T *value,
            CF::BooleanType release = 0
        );
        ~Sequence();

        Sequence(const Sequence&);
        Sequence &operator=(const Sequence&);

        T &operator[](CF::ULongType index);
        const T &operator[](CF::ULongType index) const;

        CF::ULongType length() const;
        void length(CF::ULongType);
        CF::ULongType maximum() const;

        CF::BooleanType release() const;

        void replace(
            CF::ULongType max,
            CF::ULongType length,
            T *data,
            CF::BooleanType release = 0
        );

        const T* get_buffer() const;
        T* get_buffer(CF::BooleanType orphan = 0);

        static T *allocbuf(CF::ULongType nelems);
        static void freebuf(T *);
    };
}
```

```
template <class T>
class Sequence_String : public Sequence<T>
{
    public:
    const T operator[] (CF::ULongType index) const;
};
}

#endif //__CFSEQUENCETEMPLATES_DEFINED
```