

```
/*
** 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 C: Core Framework Interface Description Language (IDL)
** Version: 4.1, 20 August 2015
*/

//Source file: CFFFile.idl

#ifndef __CFFILE_DEFINED
#define __CFFILE_DEFINED

#include "CFCommonTypes.idl"

module CF {

    /* This exception indicates a file-related error occurred.
    The message provides information describing the error. */
    exception FileException {
        CF::ErrorNumberType errorNumber;
        string msg;
    };

    /* This interface provides the ability to read and write files
    residing within a distributed FileSystem. A file can be thought of
    conceptually as a sequence of octets with a current filePointer
    describing where the next read or write will occur. */
    interface File {

        /* This exception indicates an error occurred during a read
        or write operation to a File. The message is component-dependent,
        providing additional information describing the reason for
        the error. */
        exception IOException {
            CF::ErrorNumberType errorNumber;
            string msg;
        };

        /* This exception indicates the file pointer is out of range based upon
        the current file size. */
        exception InvalidFilePointer {
        };

        /* The readonly attribute contains the file name given
        to the FileSystem open/create operation. */
        readonly attribute string fileName;

        /* The readonly attribute contains the file position
        where the next read or write will occur. */
        readonly attribute unsigned long filePointer;

        /* Applications require the read operation in order to retrieve
        data from remote files. */
        void read (
```

```
        out CF::OctetSequence data,
        in unsigned long length
    )
    raises (CF::File::IOException);

/* This operation writes data to the file referenced. */
void write (
    in CF::OctetSequence data
)
    raises (CF::File::IOException);

/* This operation returns the current size of the file. */
unsigned long sizeof ()
    raises (CF::FileException);

/* This operation releases any OE file resources associated
with the component. */
void close ()
    raises (CF::FileException);

/* This operation positions the file pointer where
next read or write will occur. */
void setFilePointer (
    in unsigned long filePointer
)
    raises (CF::File::InvalidFilePointer, CF::FileException);
};
#endif
```