

```
/*
** 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: CFFFileManager.idl

#ifndef __CFFILEMANAGER_DEFINED
#define __CFFILEMANAGER_DEFINED

#include "CFFFileSystem.idl"

module CF {

    /* Multiple, distributed FileSystems may be accessed through
       a FileManager. The FileManager interface appears to be a single
       FileSystem although the actual file storage may span multiple
       physical file systems. */
    interface FileManager : FileSystem {

        /* This structure identifies the FileSystems mounted within
           the FileManager. */
        struct MountType {
            string mountPoint;
            CF::FileSystem fs;
        };

        /* This type defines an unbounded sequence of mounted FileSystems. */
        typedef sequence <MountType> MountSequence;

        /* This exception indicates a mount point does not exist within
           the FileManager. */
        exception NonExistentMount {
        };

        /* This exception indicates the FileSystem is a null (nil) object
           reference. */
        exception InvalidFileSystem {
        };

        /* This exception indicates the mount point is already in
           use in the FileManager. */
        exception MountPointAlreadyExists {
        };

        /* This operation associates a FileSystem with a mount point
           (a directory name). */
        void mount (
            in string mountPoint,
            in CF::FileSystem file_System
        )
        raises (CF::InvalidFileName,
```

```
        CF::FileManager::InvalidFileSystem,
        CF::FileManager::MountPointAlreadyExists);

/* This operation removes a mounted FileSystem from
   the FileManager whose mounted name matches the input mountPoint
   name. */
void unmount (
    in string mountPoint
)
    raises (CF::FileManager::NonExistentMount);

/* This operation returns the FileManager's mounted
   FileSystems. */
CF::FileManager::MountSequence getMounts ();

};
#endif
```