

```
/*
** 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: CFPortAccessor.idl

#ifndef __CFPORTACCESSOR_DEFINED
#define __CFPORTACCESSOR_DEFINED

module CF {

    interface PortAccessor {

        /* This structure defines a type for information needed to disconnect a
        connection. */
        struct ConnectionIdType {
            string connectionId;
            string portName;
        };

        /* The sequence of ConnectionIdType structures. */
        typedef sequence <ConnectionIdType> Disconnections;

        /* This structure defines a type for information needed to make a
        connection. */
        struct ConnectionType {
            ConnectionIdType portConnectionId;
            Object portReference;
        };

        /* This type defines a sequence of ConnectionType structures. */
        typedef sequence <ConnectionType> Connections;

        /* This structure identifies a port and associated error code
        to be provided in the InvalidPort exception. */
        struct ConnectionErrorType {
            ConnectionIdType portConnectionId;
            unsigned short errorCode;
        };

        /* This exception indicates one of the following errors has occurred in
        the specification of a PortAccessor association. */
        exception InvalidPort {
            ConnectionErrorType invalidConnections;
        };

        /* This operation supplies a component with a sequence of
        connection information. */
        void connectUsesPorts(
            in CF::PortAccessor::Connections portConnections)
            raises(CF::PortAccessor::InvalidPort);
    };
};

#endif
```

```
/* This operation releases a sequence of uses or
   provides ports from a given connection(s). */
void disconnectPorts(
    in CF::PortAccessor::Disconnections portDisconnections)
    raises(CF::PortAccessor::InvalidPort );

/* This operation provides a mechanism to
   obtain a specific provides port(s). */
void getProvidesPorts(
    inout CF::PortAccessor::Connections portConnections )
    raises(CF::PortAccessor::InvalidPort);
};

};
#endif
```