

```
/*
** 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: CFCapacityManagement.idl

#ifndef __CFCAPACITYMANAGEMENT_DEFINED
#define __CFCAPACITYMANAGEMENT_DEFINED

#include "CFProperties.idl"
#include "CFPlatformTypes.idl"

module CF {

    /* This interface defines additional capabilities and an
       attribute for any logical device in the domain. */
    interface CapacityManagement {

        /* This enumeration type defines the device's usage states. */
        enum UsageType {
            IDLE,
            ACTIVE,
            BUSY
        };

        /* This readonly attribute contains the device's usage
           state. The usageState indicates whether or not a device is
           actively in use at a specific instant, and if so, whether
           or not it has spare capacity for allocation at that instant. */
        readonly attribute CF::CapacityManagement::UsageType usageState;

        /* This exception returns the capacities that are
           not valid for this device. */
        exception InvalidCapacity {

            /* The message indicates the reason for the invalid capacity. */
            string msg;

            /* The invalid capacities sent to the allocateCapacity operation. */
            CF::Properties capacities;
        };

        /* This operation provides the mechanism to request
           and allocate capacity from the device. */
        boolean allocateCapacity (
            in CF::Properties capacities
        )
        raises (CF::CapacityManagement::InvalidCapacity,
            CF::InvalidState);

        /* This operation provides the mechanism to return
```

```
        capacities back to the device, making them available to other
        users. */
void deallocateCapacity (
    in CF::Properties capacities
)
    raises (CF::CapacityManagement::InvalidCapacity,
           CF::InvalidState);
};
#endif
```