```
/*
** 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: CFExecutableInterface.idl


#ifndef __CFEXECUTABLEINTERFACE_DEFINED
#define __CFEXECUTABLEINTERFACE_DEFINED

#include "CFPlatformTypes.idl"

module CF {

      /* This interface defines execute and terminate behavior to a device. */
      interface ExecutableInterface

      {

      /* This exception indicates that a process,
         as identified by the processId parameter, does not exist on this
         device. The message provides additional information describing
         the reason for the error. */
      exception InvalidProcess {
         CF::ErrorNumberType errorNumber;
         string msg;
      };

      /* This exception indicates that a function, as identified by
         the input name parameter, hasn't been loaded on this device. */
      exception InvalidFunction {
      };

      /* This type defines a structure to hold the process number or thread id
         within the system. The number is unique to the Processor operating system
         that created the process/thread. */
      struct ExecutionID_Type {
         unsigned long long threadId;
         unsigned long long processId;
         string processCollocation;
         CF::ULongSeq cores;
      };

      /* This exception indicates that input parameters
         are invalid for the execute operation. Each parameter's ID and
         value must be a valid string type. The invalidParms is a list
         of invalid parameters specified in the execute operation. */
      exception InvalidParameters {
         CF::Properties invalidParms;
      };
```

```
      /* This exception indicates the input options are
         invalid on the execute operation. The invalidOptions is a list
         of invalid options specified in the execute operation. */
      exception InvalidOptions {
         CF::Properties invalidOpts;
      };

      /* The STACK_SIZE_ID is the identifier for the ExecutableInterface's
         execute options parameter. */
      const string STACK_SIZE_ID = "STACK_SIZE";

       /* The PRIORITY_ID is the identifier for the ExecutableInterface's
         execute options parameters. */
      const string PRIORITY_ID = "PRIORITY";

       /* The EXEC_DEVICE_PROCESS_SPACE is the identifier for the ExecutableInterface's
         execute options PROCESS_COLLOCATION_ID parameter. */
      const string EXEC_DEVICE_PROCESS_SPACE = "DEVICE";

       /* The PROCESS_COLLOCATION_ID is the identifier for the ExecutableInterface's
         execute options PROCESS_COLLOCATION_ID parameter. */
      const string PROCESS_COLLOCATION_ID = "PROCESS_COLLOCATION";

       /* The ENTRY_POINT_ID is the identifier for the ExecutableInterface's
         execute options parameters. */
      const string ENTRY_POINT_ID = "ENTRY_POINT";

     /* The CORE_AFFINITY_ID is the identifier for the ExecutableInterface's
         execute options parameters. */
      const string CORE_AFFINITY_ID = "CORE_AFFINITY";

      /* This exception indicates that an attempt to invoke
         the execute operation on a device failed. The message provides
         additional information describing the reason for the error. */
      exception ExecuteFail {
         CF::ErrorNumberType errorNumber;
         string msg;
      };

      /* This operation provides the mechanism for terminating
         the execution of a process/thread on a specific device that was
         started up with the execute operation. */
      void terminate (
            in CF::ExecutableInterface::ExecutionID_Type executionId
            )
         raises (CF::ExecutableInterface::InvalidProcess,
                 CF::InvalidState);

      /* This operation provides the mechanism for starting up and
         executing a software process/thread on a device. */
      CF::ExecutableInterface::ExecutionID_Type execute (
            in string filename,
            in CF::Properties options,
            in CF::Properties parameters
            )
         raises (CF::InvalidState,
                 CF::ExecutableInterface::InvalidFunction,
                 CF::ExecutableInterface::InvalidParameters,
                 CF::ExecutableInterface::InvalidOptions,
                 CF::InvalidFileName,
                 CF::ExecutableInterface::ExecuteFail);
   };
};
#endif
```