

```
/*
** 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: SE_StateEvent.idl

#ifndef __SE_STATEEVENT_DEFINED
#define __SE_STATEEVENT_DEFINED

module StandardEvent {

    /* This enumeration is utilized
       in the StateChangeEvent. It is used to identify the category
       of state change that has occurred. */
    enum StateChangeCategoryType {
        ADMINISTRATIVE_STATE_EVENT,
        OPERATIONAL_STATE_EVENT,
        USAGE_STATE_EVENT
    };

    /* This enumeration is utilized
       in the StateChangeEvent. It is used to identify the specific
       states of the event source before and after the state change
       occurred. */
    enum StateChangeType {
        LOCKED,
        UNLOCKED,
        SHUTTING_DOWN,
        ENABLED,
        DISABLED,
        IDLE,
        ACTIVE,
        BUSY
    };

    /* This structure is used to indicate that
       the state of the event source has changed. The event producer
       will send this structure into an event channel on behalf of
       the event source. */
    struct StateChangeEvent {
        string producerId;
        string sourceId;
        StandardEvent::StateChangeCategoryType stateChangeCategory;
        StandardEvent::StateChangeType stateChangeFrom;
        StandardEvent::StateChangeType stateChangeTo;
    };

};

#endif
```