```
#ifndef _EDU_WSU_GRIDSTAT_COMMAND_INTERFACES
#define _EDU_WSU_GRIDSTAT_COMMAND_INTERFACES

#include "commStructs.idl"

module edu
{
module wsu
{
  module gridstat
  {
    module command
    {
      ////////////////////////////////////////////////////////////////
      // Commands that can be issued by the leaf QoS broker to edge
      // status routers and status routers
      interface CommandLeafToSRCommon
      {
        // Method to connection a event channel
        short connectLink(in EventChannelInfo eventChannel);

        // Method to disconnect a event channel
        short disconnectLink(in string dstSRName);

        // Method to establish a path in the routing table
        short setupPath(in unsigned long variableId,
                        in string dstSRName,
                        in unsigned long pathId,
                        in unsigned short redundantId,
                        in IntervalSeq subInterval,
                        in unsigned long pubInterval,
                        in PrioritySeq priority,
                        in unsigned long modes);

        // Method to terminate a paths in the routing table
        short terminatePath(in unsigned long variableId,
                            in string dstSRName,
                            in unsigned long pathId,
                            in unsigned short redundantId,
                            in IntervalSeq subInterval,
                            in unsigned long modes);

        // Method to change the priority of an established path
        short changePathPriority(in unsigned long variableId,
                                 in string dstSRName,
                                 in unsigned long pathId,
                                 in unsigned short redundantId,
                                 in unsigned long subInterval,
                                 in unsigned long modes,
                                 in unsigned short newPriority);

        // Method to change the interval of an established path
        short changePathSubInterval(in unsigned long variableId,
                                    in string dstSRName,
                                    in unsigned long pathId,
                                    in unsigned short redundantId,
                                    in unsigned long subInterval,
                                    in unsigned long modes,
                                    in unsigned long newSubInterval);

        // Method to change the pub interval of a variable that has
        // at least one path through this SR
        short changeVarPubInterval(in unsigned long variableId,
                                   in unsigned long newPubInterval);

        // Method to create a condensation function
        short setupCondensation(
                              in CondensationSetupInfo condHolder,
                              in unsigned long variableId);

        // Method to remove an condensation function
        short terminateCondensation(in unsigned long variableId);

        // Method to set one of the variables to be flooded
        short setupFlooding(in unsigned long variableId,
                            in unsigned long pubInterval,
                            in unsigned long level,
                            in unsigned long modes);

        // Method to terminate the flooding of one of the variables
        short terminateFlooding(in unsigned long variableId,
                                in unsigned long modes);
      };

      ////////////////////////////////////////////////////////////////
      // Commands that can be issued by the leaf QoS broker
      // to the status routers
      interface CommandLeafToSR : CommandLeafToSRCommon
      {
        // Method to set the new mode
        short changeMode(in unsigned long mode);
      };

      ////////////////////////////////////////////////////////////////
      // Commands that can be issued by the leaf QoS broker to the
      // edge status routers
      interface CommandLeafToESR : CommandLeafToSRCommon
      {
        // Method to inform about a mode change
        short informChangeMode(in unsigned long mode);

        // Method to prepare for a mode change
        short prepareChangeMode(in unsigned long mode);

        // Method to set the new mode
        short changeMode(in unsigned long mode);

        // Method to commit a mode change
        short commitChangeMode(in unsigned long mode);

        // Method to forward an error msg that a subscription failed
        short subscribeErrorMsg(in string subName,
                                in string pubName,
                                in string variableName,
                                in unsigned short errorCode);
      };

      ////////////////////////////////////////////////////////////////
      // Commands that can be issued by publishers to its
      // leaf QoS broker
      interface CommandPubToESR
      {
        // Method to register publishing of a variable
        short publish(in PublishInfo pubHolder,
                      inout unsigned long variableId);

        // Method to unregister a publishing of a variable
        short unPublish(in string pubName, in string variableName);

        // Method to register a publisher
        short registerPublisher(in string pubName,
                                inout string hostName,
                                inout unsigned long hostPort,
                                in boolean firstTime);

        // Method to unregister a publisher
        short unRegisterPublisher(in string pubName);

        // Used to check if the edge status router is running
        boolean pubPing();
      };

      ////////////////////////////////////////////////////////////////
      // Commands that can be issued by subscribers to its
      // leaf QoS broker
      interface CommandSubToESR
      {
        // Method to register a subscription
        short subscribe(in SubscribeInfo subHolder,
                        out unsigned long variableId,
                        out unsigned short numPatterns);

        // Method to unregister a subscription
        short unSubscribe(in string pubName,
                          in string variableName,
                          in string subName);

        // Method to register a subscriber
        short registerSubscriber(in string subName,
                                 inout string hostName,
                                 inout unsigned long hostPort,
                                 out unsigned long currentMode,
                                 in boolean firstTime);

        // Method to unregister a subscriber
        short unRegisterSubscriber(in string subName);

        // Used to check if the edge status router is running
        boolean subPing();
      };
```