

smacc::ISmaccClientBehavior
<ul style="list-style-type: none"> - stateMachine_ - currentState - currentOrthogonal
<ul style="list-style-type: none"> + ISmaccClientBehavior() + ~ISmaccClientBehavior() + getStateMachine() + getName() + requiresClient() + requiresComponent() + getNode() # runtimeConfigure() # onEntry() # onExit() # postEvent() # postEvent() # getCurrentState() # executeOnEntry() # executeOnExit() # dispose() - onOrthogonalAllocation()

smacc::SmaccAsyncClientBehavior
<ul style="list-style-type: none"> - onEntryThread_ - onExitThread_ - postFinishEventFn_ - postSuccessEventFn_ - postFailureEventFn_ - onFinished_ - onSuccess_ - onFailure_
<ul style="list-style-type: none"> + onOrthogonalAllocation() + ~SmaccAsyncClientBehavior() + onSuccess() + onFinished() + onFailure() # executeOnEntry() # executeOnExit() # postSuccessEvent() # postFailureEvent() # dispose()

smacc::ISmaccUpdatable
<ul style="list-style-type: none"> - periodDuration_ - lastUpdate_
<ul style="list-style-type: none"> + ISmaccUpdatable() + ISmaccUpdatable() + executeUpdate() + setUpdatePeriod() # update()

cl_move_group_interface ::CbMoveEndEffectorTrajectory
<ul style="list-style-type: none"> + group_ + tipLink_ + allowInitialTrajectoryState JointDiscontinuity_ # endEffectorTrajectory_ # movegroupClient_ # beahiorMarkers_ - markersPub_ - markersIntialized_ - iksrv_ - m_mutex_ - postJointDiscontinuityEvent - postIncorrectInitialState Event - postMotionExecutionFailure Events - autocleanmarkers
<ul style="list-style-type: none"> + CbMoveEndEffectorTrajectory() + CbMoveEndEffectorTrajectory() + onOrthogonalAllocation() + onEntry() + onExit() + update() # computeJointSpaceTrajectory() # executeJointSpaceTrajectory() # generateTrajectory() # createMarkers() # getCurrentEndEffectorPose() - initializeROS()

cl_move_group_interface ::CbMoveCartesianRelative2
<ul style="list-style-type: none"> + offset_ + linearSpeed_m_s_ - globalFrame_
<ul style="list-style-type: none"> + CbMoveCartesianRelative2() + CbMoveCartesianRelative2() + generateTrajectory()