{shared_ptr< smacc2 ::introspection::SmaccState MachineInfo > ||}
{shared_ptr< smacc2 ::introspection::SmaccState Info > ||}
{SharedPtr ||}
{uint64_t ||}
{vector< smacc2::ISmaccState * > ||}
{SmaccStatus ||}
{map< std::string, std ::shared_ptr< smacc2 ::ISmaccOrthogonal > > ||}
{vector< boost::signals2 ::scoped_connection > ||}
{SharedPtr ||}
{recursive_mutex ||}
{StateMachineInternalAction ||}
{map< void *, std::shared _ptr< smacc2::CallbackCounterSemaphore > > ||}
{map< std::string, std ::pair< std::function < std::string()>, boost ::any > > ||}
{vector< smacc2_msgs ::msg::SmaccTransitionLogEntry > ||}
{SMRunMode ||}
{SharedPtr ||}
#stateMachineInfo_
#currentStateInfo_
#nh_
-stateSeqCounter_
#currentState_
#status_msg_
#orthogonals_
#longLivedSignalConnections_
#timer_
-eventQueueMutex_ -m_mutex_
-stateMachineCurrentAction
-stateCallbackConnections
-globalData_
-transitionLogHistory_
-runMode_
#stateMachinePub_ #stateMachineStatusPub _ #transitionHistoryService _ #transitionLogPub_
{smacc2::ISmaccStateMachine ||+ ISmaccStateMachine() + ~ISmaccStateMachine() + reset() + stop() + eStop() + getOrthogonal() + getClientBehavior() + getOrthogonals() + requiresComponent() + postEvent() + postEvent() + getGlobalSMData() + setGlobalSMData() + mapBehavior() + getStateMachineName() + state_machine_visualization() + getCurrentStateInfo() + publishTransition() + onInitialize() + getTransitionLogHistory() + createSignalConnection() + disconnectSmaccSignalObject() + notifyOnStateEntryStart() + notifyOnStateEntryEnd() + notifyOnRuntimeConfigured() + notifyOnStateExitting() + notifyOnStateExited() + notifyOnRuntimeConfiguration Finished() + getCurrentStateCounter() + getCurrentState() + getStateMachineInfo() + buildStateMachineInfo() + getNode() + getLogger() + getMutex() # checkStateMachineConsistence() # initializeROS() # onInitialized() # createOrthogonal() - lockStateMachine() - unlockStateMachine() - propagateEventToStateReactors() - updateStatusMessage() }
{type_info * ||}
+stateMachine_
+parentState_
{map< const std::type _info *, std::vector < smacc2::introspection ::ClientBehaviorInfoEntry > > ||}
{map< const std::type _info *, std::vector < std::shared_ptr< smacc2 ::introspection::SmaccStateReactor Info > > > ||}
{map< const std::type _info *, std::vector < std::shared_ptr< smacc2 ::introspection::SmaccEventGenerator Info > > > ||}
{int ||}
{vector< smacc2::introspection ::SmaccTransitionInfo > ||}
{std::enable_shared _from_this< SmaccStateInfo > ||}
{vector< std::shared _ptr< smacc2::introspection ::SmaccStateInfo > > ||}
{string ||}
{vector< std::shared _ptr< smacc2::ISmaccClient > > ||}
{vector< std::vector < std::shared_ptr< smacc2 ::ISmaccClientBehavior > > > ||}
{mutex ||}
{double ||}
-statusPub_
{vector< smacc2::ISmaccUpdatable * > ||}
{vector< std::vector < smacc2::ISmaccUpdatable * > > ||}
{atomic< uint64_t > ||}
{atomic< bool > ||}
{SmaccFifoScheduler * ||}
{processor_handle ||}
{thread ||}
{ExecutionModel ||}
+tid_
+staticBehaviorInfo
+stateReactorsInfo
+eventGeneratorsInfo
+depth_ +stateIndex_
+transitions_
+children_
#node_
+demangledStateName +fullStateName
#clients_
-clientBehaviors_
-mutex_
-stateMachine_
-signalDetector_
-smaccStateMachine_
-loop_rate_hz
-updatableClients_
-updatableStateElements_
-lastState_
-end_ -initialized_
-scheduler_
-processorHandle_
-signalDetectorThread_
-executionModel_
{smacc2::introspection ::SmaccStateInfo ||+ SmaccStateInfo() + getStateLevel() + depth() + getAncestors() + getFullPath() + createChildState() + declareTransition() + toShortName() + getDemangledFullName() + getNode() + getLogger() }
{shared_ptr< rclcpp ::Logger > ||}
{vector< std::shared _ptr< smacc2::StateReactor > > ||}
{vector< std::shared _ptr< smacc2::SmaccEventGenerator > > ||}
{boost::signals2::signal < Signature, optional _last_value< typename boost ::function_traits< Signature >::result_type >, int, std ::less< int >, function< Signature >, typename extended_signature< function_traits< Signature >::arity, Signature >::function_type, boost:: signals2::mutex > ||}
{smacc2::ISmaccOrthogonal ||+ setStateMachine() + getStateMachine() + initState() + addClientBehavior() + runtimeConfigure() + onEntry() + onExit() + onDispose() + getName() + requiresComponent() + requiresClient() + getClients() + getClientBehaviors() + setGlobalSMData() + getGlobalSMData() + getClientBehavior() + getNode() + getLogger() # onInitialize() # initializeClients() # assignClientToOrthogonal() }
{map< smacc2::ComponentKey, std::shared_ptr< smacc2 ::ISmaccComponent > > ||}
-stateMachine_
{smacc2::SignalDetector ||+ SignalDetector() + initialize() + setProcessorHandle() + runThread() + join() + stop() + pollingLoop() + postEvent() + getNode() + getLogger() + notifyStateConfigured() + notifyStateExited() - pollOnce() - findUpdatableClientsAndComponents() - findUpdatableStateElements() }
#stateInfo_
#logger_
#stateReactors_
#eventGenerators_
#name_
-orthogonal_
#components_
-stateMachine_
#stateMachine_
#odomSub_ #robotBasePathPub _ #robotBasePathStacked Pub_
{smacc2::ISmaccState ||+ getStateMachine() + getName() + getParentState() + getNode() + getLogger() + getClassName() + configure() + requiresComponent() + requiresClient() + getGlobalSMData() + setGlobalSMData() + createStateReactor() + createStateReactor() + createEventGenerator() + postEvent() + postEvent() + notifyTransition() + notifyTransitionFromTransition TypeInfo() + getStateReactors() + getEventGenerators() + getOrthogonal() + getClientBehavior() + getEventGenerator() + getStateReactor() }
#parentState_
-currentOrthogonal
{boost::signals2::signal < void(), optional_last _value< typename boost:: function_traits< void() > ::result_type >, int, std ::less< int >, function< void () >, typename extended_signature < function_traits< void() >::arity, void() >::function_type, boost::signals2::mutex > ||}
{smacc2::SmaccSignal < Signature, Combiner, Group, GroupCompare, SlotFunction, ExtendedSlotFunction, Mutex > ||}
{boost::signals2::signal < void(const WrappedResult &), optional_last_value < typename boost::function _traits< void(const WrappedResult &) >::result_type >, int, std ::less< int >, function< void(const WrappedResult &) >, typename extended _signature< function_traits< void(const WrappedResult &) >::arity, void(const WrappedResult &) >::function_type, boost::signals2::mutex > ||}
+action_endpoint_
#currentPathName_ #odomFrame_ #odomTopicName_
#m_mutex_
{smacc2::ISmaccClient ||+ ISmaccClient() + ~ISmaccClient() + onInitialize() + getName() + getComponent() + getComponent() + getComponent() + getType() + getStateMachine() + connectSignal() + requiresClient() + getComponents() + iterateComponents() + postEvent() + postEvent() # onOrthogonalAllocation() # createComponent() # createNamedComponent() # getNode() # getLogger() - initialize() - setStateMachine() - setOrthogonal() }
#clearAngularDistanceThreshold _ #clearPointDistanceThreshold _ #recordAngularDistanceThreshold _ #recordPointDistanceThreshold_
-currentState
< void()>
< void(const WrappedResult &)>
+action_endpoint_
#owner_
{smacc2::ISmaccClientBehavior ||+ ISmaccClientBehavior() + ~ISmaccClientBehavior() + getStateMachine() + getName() + requiresClient() + requiresComponent() + onEntry() + onExit() + executeOnEntry() + executeOnExit() # runtimeConfigure() # postEvent() # postEvent() # getCurrentState() # dispose() # getNode() # getLogger() - onOrthogonalAllocation() }
{smacc2::SmaccSignal < void()> ||}
{optional< std::future < int > > ||}
{function< void()> ||}
{SharedPtr ||}
{smacc2::SmaccSignal < void(const WrappedResult &)> ||}
{FeedbackCallback ||}
{smacc2::client_bases ::ISmaccActionClient ||+ ISmaccActionClient() + ~ISmaccActionClient() + getNamespace() + cancelGoal() + getClientBase() + getName() }
{bool ||}
{function< void(WrappedResult)> ||}
{optional< std::shared _future< typename GoalHandle ::SharedPtr > > ||}
{function< void(const Feedback &)> ||}
{Path ||}
{WorkingMode ||}
{vector< nav_msgs::msg ::Path > ||}
{vector< cl_nav2z::odom _tracker::CpOdomTracker ::PathInfo > ||}
{optional< geometry _msgs::msg::PoseStamped > ||}
{smacc2::ISmaccComponent ||+ ISmaccComponent() + ~ISmaccComponent() + getName() # onInitialize() # postEvent() # postEvent() # onOrthogonalAllocation() # requiresComponent() # requiresComponent() # requiresClient() # createSiblingComponent() # createSiblingNamedComponent() # getNode() # getLogger() # getStateMachine() - initialize() - setStateMachine() }
-onFailure_ -onFinished_ -onSuccess_
-onEntryThread_ -onExitThread_
-postFailureEventFn _ -postFinishEventFn _ -postSuccessEventFn_
#client_
+onAborted_ +onCancelled_ +onSucceeded_
-isShutdownRequested_
+feedback_cb
#publishMessages #subscribeToOdometryTopic_
+postAbortedEvent +postCancelledEvent +postSuccessEvent
+lastRequest_
+postFeedbackEvent
#aggregatedStackPathMsg _ #baseTrajectory_
#workingMode_
#pathStack_
#pathInfos_
#currentMotionGoal_
{smacc2::SmaccAsyncClient Behavior ||+ onOrthogonalAllocation() + ~SmaccAsyncClientBehavior() + onSuccess() + onFinished() + onFailure() + requestForceFinish() + executeOnEntry() + executeOnExit() + waitOnEntryThread() + onSuccess() + onFinished() + onFailure() # postSuccessEvent() # postFailureEvent() # dispose() # isShutdownRequested() - waitFutureIfNotFinished() }
{smacc2::client_bases ::SmaccActionClientBase < ActionType > ||+ SmaccActionClientBase() + SmaccActionClientBase() + ~SmaccActionClientBase() + getClientBase() + onInitialize() + postResultEvent() + onOrthogonalAllocation() + onSucceeded() + onSucceeded() + onAborted() + onAborted() + onCancelled() + onCancelled() + cancelGoal() + sendGoal() + getEventLabel() # onFeedback() # onResult() }
{cl_nav2z::odom_tracker ::CpOdomTracker ||+ CpOdomTracker() + processOdometryMessage() + setWorkingMode() + setPublishMessages() + pushPath() + pushPath() + popPath() + clearPath() + setStartPoint() + setStartPoint() + setCurrentMotionGoal() + setCurrentPathName() + getCurrentMotionGoal() + getPath() + logStateString() # onInitialize() # updateConfiguration() # rtPublishPaths() # updateRecordPath() # updateClearPath() # updateAggregatedStackPath() }
#client_
+onAborted_ +onCancelled_ +onSucceeded_
+feedback_cb
+postAbortedEvent +postCancelledEvent +postSuccessEvent
+lastRequest_
+postFeedbackEvent
< nav2_msgs::action ::NavigateToPose >
{smacc2::client_bases ::SmaccActionClientBase < nav2_msgs::action::NavigateTo Pose > ||+ SmaccActionClientBase() + SmaccActionClientBase() + ~SmaccActionClientBase() + getClientBase() + onInitialize() + postResultEvent() + onOrthogonalAllocation() + onSucceeded() + onSucceeded() + onAborted() + onAborted() + onCancelled() + onCancelled() + cancelGoal() + sendGoal() + getEventLabel() # onFeedback() # onResult() }
-odomTracker
{cl_nav2z::ClNav2Z ||+ ClNav2Z() + ~ClNav2Z() }
{SharedPtr ||}
{ResultCode ||}
{shared_future< std ::shared_ptr< rclcpp _action::ClientGoalHandle < nav2_msgs::action::NavigateTo Pose > > > ||}
#nav2zClient_
#navigationCallback_
#navigationResult_
#goalHandleFuture_
{cl_nav2z::CbNav2ZClientBehaviorBase ||+ ~CbNav2ZClientBehaviorBase() + onOrthogonalAllocation() # sendGoal() # cancelGoal() # isOwnActionResponse() # onNavigationResult() # onNavigationActionSuccess() # onNavigationActionAbort() }
{shared_ptr< tf2_ros ::Buffer > ||}
{optional< cl_nav2z ::CbUndoPathBackwardsOptions > ||}
-listener
-options_
{cl_nav2z::CbUndoPathBackwards ||+ CbUndoPathBackwards() + onEntry() + onExit() }