;; Auto-generated. Do not edit!

(when (boundp 'gazebo\_msgs\_new::GetLightProperties)

(if (not (find-package "GAZEBO\_MSGS\_NEW"))

(make-package "GAZEBO\_MSGS\_NEW"))

(shadow 'GetLightProperties (find-package "GAZEBO\_MSGS\_NEW")))

(unless (find-package "GAZEBO\_MSGS\_NEW::GETLIGHTPROPERTIES")

(make-package "GAZEBO\_MSGS\_NEW::GETLIGHTPROPERTIES"))

(unless (find-package "GAZEBO\_MSGS\_NEW::GETLIGHTPROPERTIESREQUEST")

(make-package "GAZEBO\_MSGS\_NEW::GETLIGHTPROPERTIESREQUEST"))

(unless (find-package "GAZEBO\_MSGS\_NEW::GETLIGHTPROPERTIESRESPONSE")

(make-package "GAZEBO\_MSGS\_NEW::GETLIGHTPROPERTIESRESPONSE"))

(in-package "ROS")

(if (not (find-package "STD\_MSGS"))

(ros::roseus-add-msgs "std\_msgs"))

(defclass gazebo\_msgs\_new::GetLightPropertiesRequest

:super ros::object

:slots (\_light\_name ))

(defmethod gazebo\_msgs\_new::GetLightPropertiesRequest

(:init

(&key

((:light\_name \_\_light\_name) "")

)

(send-super :init)

(setq \_light\_name (string \_\_light\_name))

self)

(:light\_name

(&optional \_\_light\_name)

(if \_\_light\_name (setq \_light\_name \_\_light\_name)) \_light\_name)

(:serialization-length

()

(+

;; string \_light\_name

4 (length \_light\_name)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

(make-string-output-stream (send self :serialization-length)))))

;; string \_light\_name

(write-long (length \_light\_name) s) (princ \_light\_name s)

;;

(if (null strm) (get-output-stream-string s))))

(:deserialize

(buf &optional (ptr- 0))

;; string \_light\_name

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setq \_light\_name (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

;;

self)

)

(defclass gazebo\_msgs\_new::GetLightPropertiesResponse

:super ros::object

:slots (\_diffuse \_attenuation\_constant \_attenuation\_linear \_attenuation\_quadratic \_success \_status\_message ))

(defmethod gazebo\_msgs\_new::GetLightPropertiesResponse

(:init

(&key

((:diffuse \_\_diffuse) (instance std\_msgs::ColorRGBA :init))

((:attenuation\_constant \_\_attenuation\_constant) 0.0)

((:attenuation\_linear \_\_attenuation\_linear) 0.0)

((:attenuation\_quadratic \_\_attenuation\_quadratic) 0.0)

((:success \_\_success) nil)

((:status\_message \_\_status\_message) "")

)

(send-super :init)

(setq \_diffuse \_\_diffuse)

(setq \_attenuation\_constant (float \_\_attenuation\_constant))

(setq \_attenuation\_linear (float \_\_attenuation\_linear))

(setq \_attenuation\_quadratic (float \_\_attenuation\_quadratic))

(setq \_success \_\_success)

(setq \_status\_message (string \_\_status\_message))

self)

(:diffuse

(&rest \_\_diffuse)

(if (keywordp (car \_\_diffuse))

(send\* \_diffuse \_\_diffuse)

(progn

(if \_\_diffuse (setq \_diffuse (car \_\_diffuse)))

\_diffuse)))

(:attenuation\_constant

(&optional \_\_attenuation\_constant)

(if \_\_attenuation\_constant (setq \_attenuation\_constant \_\_attenuation\_constant)) \_attenuation\_constant)

(:attenuation\_linear

(&optional \_\_attenuation\_linear)

(if \_\_attenuation\_linear (setq \_attenuation\_linear \_\_attenuation\_linear)) \_attenuation\_linear)

(:attenuation\_quadratic

(&optional \_\_attenuation\_quadratic)

(if \_\_attenuation\_quadratic (setq \_attenuation\_quadratic \_\_attenuation\_quadratic)) \_attenuation\_quadratic)

(:success

(&optional \_\_success)

(if \_\_success (setq \_success \_\_success)) \_success)

(:status\_message

(&optional \_\_status\_message)

(if \_\_status\_message (setq \_status\_message \_\_status\_message)) \_status\_message)

(:serialization-length

()

(+

;; std\_msgs/ColorRGBA \_diffuse

(send \_diffuse :serialization-length)

;; float64 \_attenuation\_constant

8

;; float64 \_attenuation\_linear

8

;; float64 \_attenuation\_quadratic

8

;; bool \_success

1

;; string \_status\_message

4 (length \_status\_message)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

(make-string-output-stream (send self :serialization-length)))))

;; std\_msgs/ColorRGBA \_diffuse

(send \_diffuse :serialize s)

;; float64 \_attenuation\_constant

(sys::poke \_attenuation\_constant (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; float64 \_attenuation\_linear

(sys::poke \_attenuation\_linear (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; float64 \_attenuation\_quadratic

(sys::poke \_attenuation\_quadratic (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; bool \_success

(if \_success (write-byte -1 s) (write-byte 0 s))

;; string \_status\_message

(write-long (length \_status\_message) s) (princ \_status\_message s)

;;

(if (null strm) (get-output-stream-string s))))

(:deserialize

(buf &optional (ptr- 0))

;; std\_msgs/ColorRGBA \_diffuse

(send \_diffuse :deserialize buf ptr-) (incf ptr- (send \_diffuse :serialization-length))

;; float64 \_attenuation\_constant

(setq \_attenuation\_constant (sys::peek buf ptr- :double)) (incf ptr- 8)

;; float64 \_attenuation\_linear

(setq \_attenuation\_linear (sys::peek buf ptr- :double)) (incf ptr- 8)

;; float64 \_attenuation\_quadratic

(setq \_attenuation\_quadratic (sys::peek buf ptr- :double)) (incf ptr- 8)

;; bool \_success

(setq \_success (not (= 0 (sys::peek buf ptr- :char)))) (incf ptr- 1)

;; string \_status\_message

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setq \_status\_message (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

;;

self)

)

(defclass gazebo\_msgs\_new::GetLightProperties

:super ros::object

:slots ())

(setf (get gazebo\_msgs\_new::GetLightProperties :md5sum-) "df2cef87e13e2e6990e81e8aaa454c19")

(setf (get gazebo\_msgs\_new::GetLightProperties :datatype-) "gazebo\_msgs\_new/GetLightProperties")

(setf (get gazebo\_msgs\_new::GetLightProperties :request) gazebo\_msgs\_new::GetLightPropertiesRequest)

(setf (get gazebo\_msgs\_new::GetLightProperties :response) gazebo\_msgs\_new::GetLightPropertiesResponse)

(defmethod gazebo\_msgs\_new::GetLightPropertiesRequest

(:response () (instance gazebo\_msgs\_new::GetLightPropertiesResponse :init)))

(setf (get gazebo\_msgs\_new::GetLightPropertiesRequest :md5sum-) "df2cef87e13e2e6990e81e8aaa454c19")

(setf (get gazebo\_msgs\_new::GetLightPropertiesRequest :datatype-) "gazebo\_msgs\_new/GetLightPropertiesRequest")

(setf (get gazebo\_msgs\_new::GetLightPropertiesRequest :definition-)

"string light\_name # name of Gazebo Light

---

std\_msgs/ColorRGBA diffuse # diffuse color as red, green, blue, alpha

float64 attenuation\_constant

float64 attenuation\_linear

float64 attenuation\_quadratic

bool success # return true if get successful

string status\_message # comments if available

================================================================================

MSG: std\_msgs/ColorRGBA

float32 r

float32 g

float32 b

float32 a

")

(setf (get gazebo\_msgs\_new::GetLightPropertiesResponse :md5sum-) "df2cef87e13e2e6990e81e8aaa454c19")

(setf (get gazebo\_msgs\_new::GetLightPropertiesResponse :datatype-) "gazebo\_msgs\_new/GetLightPropertiesResponse")

(setf (get gazebo\_msgs\_new::GetLightPropertiesResponse :definition-)

"string light\_name # name of Gazebo Light

---

std\_msgs/ColorRGBA diffuse # diffuse color as red, green, blue, alpha

float64 attenuation\_constant

float64 attenuation\_linear

float64 attenuation\_quadratic

bool success # return true if get successful

string status\_message # comments if available

================================================================================

MSG: std\_msgs/ColorRGBA

float32 r

float32 g

float32 b

float32 a

")

(provide :gazebo\_msgs\_new/GetLightProperties "df2cef87e13e2e6990e81e8aaa454c19")