;; Auto-generated. Do not edit!

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

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

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

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

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

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

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

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

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

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

(in-package "ROS")

(defclass gazebo\_msgs\_new::GetWorldPropertiesRequest

:super ros::object

:slots ())

(defmethod gazebo\_msgs\_new::GetWorldPropertiesRequest

(:init

(&key

)

(send-super :init)

self)

(:serialization-length

()

(+

0

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;;

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

(:deserialize

(buf &optional (ptr- 0))

;;

self)

)

(defclass gazebo\_msgs\_new::GetWorldPropertiesResponse

:super ros::object

:slots (\_sim\_time \_model\_names \_rendering\_enabled \_success \_status\_message ))

(defmethod gazebo\_msgs\_new::GetWorldPropertiesResponse

(:init

(&key

((:sim\_time \_\_sim\_time) 0.0)

((:model\_names \_\_model\_names) (let (r) (dotimes (i 0) (push "" r)) r))

((:rendering\_enabled \_\_rendering\_enabled) nil)

((:success \_\_success) nil)

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

)

(send-super :init)

(setq \_sim\_time (float \_\_sim\_time))

(setq \_model\_names \_\_model\_names)

(setq \_rendering\_enabled \_\_rendering\_enabled)

(setq \_success \_\_success)

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

self)

(:sim\_time

(&optional \_\_sim\_time)

(if \_\_sim\_time (setq \_sim\_time \_\_sim\_time)) \_sim\_time)

(:model\_names

(&optional \_\_model\_names)

(if \_\_model\_names (setq \_model\_names \_\_model\_names)) \_model\_names)

(:rendering\_enabled

(&optional \_\_rendering\_enabled)

(if \_\_rendering\_enabled (setq \_rendering\_enabled \_\_rendering\_enabled)) \_rendering\_enabled)

(: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

()

(+

;; float64 \_sim\_time

8

;; string[] \_model\_names

(apply #'+ (mapcar #'(lambda (x) (+ 4 (length x))) \_model\_names)) 4

;; bool \_rendering\_enabled

1

;; 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)))))

;; float64 \_sim\_time

(sys::poke \_sim\_time (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; string[] \_model\_names

(write-long (length \_model\_names) s)

(dolist (elem \_model\_names)

(write-long (length elem) s) (princ elem s)

)

;; bool \_rendering\_enabled

(if \_rendering\_enabled (write-byte -1 s) (write-byte 0 s))

;; 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))

;; float64 \_sim\_time

(setq \_sim\_time (sys::peek buf ptr- :double)) (incf ptr- 8)

;; string[] \_model\_names

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_model\_names (make-list n))

(dotimes (i n)

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setf (elt \_model\_names i) (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

))

;; bool \_rendering\_enabled

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

;; 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::GetWorldProperties

:super ros::object

:slots ())

(setf (get gazebo\_msgs\_new::GetWorldProperties :md5sum-) "36bb0f2eccf4d8be971410c22818ba3f")

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

(setf (get gazebo\_msgs\_new::GetWorldProperties :request) gazebo\_msgs\_new::GetWorldPropertiesRequest)

(setf (get gazebo\_msgs\_new::GetWorldProperties :response) gazebo\_msgs\_new::GetWorldPropertiesResponse)

(defmethod gazebo\_msgs\_new::GetWorldPropertiesRequest

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

(setf (get gazebo\_msgs\_new::GetWorldPropertiesRequest :md5sum-) "36bb0f2eccf4d8be971410c22818ba3f")

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

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

"---

float64 sim\_time # current sim time

string[] model\_names # list of models in the world

bool rendering\_enabled # if X is used

bool success # return true if get successful

string status\_message # comments if available

")

(setf (get gazebo\_msgs\_new::GetWorldPropertiesResponse :md5sum-) "36bb0f2eccf4d8be971410c22818ba3f")

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

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

"---

float64 sim\_time # current sim time

string[] model\_names # list of models in the world

bool rendering\_enabled # if X is used

bool success # return true if get successful

string status\_message # comments if available

")

(provide :gazebo\_msgs\_new/GetWorldProperties "36bb0f2eccf4d8be971410c22818ba3f")