;; Auto-generated. Do not edit!

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

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

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

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

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

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

(in-package "ROS")

;;//! \htmlinclude ModelState.msg.html

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

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

(defclass gazebo\_msgs\_new::ModelState

:super ros::object

:slots (\_model\_name \_pose \_twist \_reference\_frame ))

(defmethod gazebo\_msgs\_new::ModelState

(:init

(&key

((:model\_name \_\_model\_name) "")

((:pose \_\_pose) (instance geometry\_msgs::Pose :init))

((:twist \_\_twist) (instance geometry\_msgs::Twist :init))

((:reference\_frame \_\_reference\_frame) "")

)

(send-super :init)

(setq \_model\_name (string \_\_model\_name))

(setq \_pose \_\_pose)

(setq \_twist \_\_twist)

(setq \_reference\_frame (string \_\_reference\_frame))

self)

(:model\_name

(&optional \_\_model\_name)

(if \_\_model\_name (setq \_model\_name \_\_model\_name)) \_model\_name)

(:pose

(&rest \_\_pose)

(if (keywordp (car \_\_pose))

(send\* \_pose \_\_pose)

(progn

(if \_\_pose (setq \_pose (car \_\_pose)))

\_pose)))

(:twist

(&rest \_\_twist)

(if (keywordp (car \_\_twist))

(send\* \_twist \_\_twist)

(progn

(if \_\_twist (setq \_twist (car \_\_twist)))

\_twist)))

(:reference\_frame

(&optional \_\_reference\_frame)

(if \_\_reference\_frame (setq \_reference\_frame \_\_reference\_frame)) \_reference\_frame)

(:serialization-length

()

(+

;; string \_model\_name

4 (length \_model\_name)

;; geometry\_msgs/Pose \_pose

(send \_pose :serialization-length)

;; geometry\_msgs/Twist \_twist

(send \_twist :serialization-length)

;; string \_reference\_frame

4 (length \_reference\_frame)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; string \_model\_name

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

;; geometry\_msgs/Pose \_pose

(send \_pose :serialize s)

;; geometry\_msgs/Twist \_twist

(send \_twist :serialize s)

;; string \_reference\_frame

(write-long (length \_reference\_frame) s) (princ \_reference\_frame s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; string \_model\_name

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

;; geometry\_msgs/Pose \_pose

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

;; geometry\_msgs/Twist \_twist

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

;; string \_reference\_frame

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

;;

self)

)

(setf (get gazebo\_msgs\_new::ModelState :md5sum-) "9330fd35f2fcd82d457e54bd54e10593")

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

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

"# Set Gazebo Model pose and twist

string model\_name # model to set state (pose and twist)

geometry\_msgs/Pose pose # desired pose in reference frame

geometry\_msgs/Twist twist # desired twist in reference frame

string reference\_frame # set pose/twist relative to the frame of this entity (Body/Model)

# leave empty or \"world\" or \"map\" defaults to world-frame

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

MSG: geometry\_msgs/Pose

# A representation of pose in free space, composed of position and orientation.

Point position

Quaternion orientation

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

MSG: geometry\_msgs/Point

# This contains the position of a point in free space

float64 x

float64 y

float64 z

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

MSG: geometry\_msgs/Quaternion

# This represents an orientation in free space in quaternion form.

float64 x

float64 y

float64 z

float64 w

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

MSG: geometry\_msgs/Twist

# This expresses velocity in free space broken into its linear and angular parts.

Vector3 linear

Vector3 angular

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

MSG: geometry\_msgs/Vector3

# This represents a vector in free space.

# It is only meant to represent a direction. Therefore, it does not

# make sense to apply a translation to it (e.g., when applying a

# generic rigid transformation to a Vector3, tf2 will only apply the

# rotation). If you want your data to be translatable too, use the

# geometry\_msgs/Point message instead.

float64 x

float64 y

float64 z

")

(provide :gazebo\_msgs\_new/ModelState "9330fd35f2fcd82d457e54bd54e10593")