;; Auto-generated. Do not edit!

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

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

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

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

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

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

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

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

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

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

(in-package "ROS")

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

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

(defclass gazebo\_msgs\_new::SpawnModelRequest

:super ros::object

:slots (\_model\_name \_model\_xml \_robot\_namespace \_initial\_pose \_reference\_frame ))

(defmethod gazebo\_msgs\_new::SpawnModelRequest

(:init

(&key

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

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

((:robot\_namespace \_\_robot\_namespace) "")

((:initial\_pose \_\_initial\_pose) (instance geometry\_msgs::Pose :init))

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

)

(send-super :init)

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

(setq \_model\_xml (string \_\_model\_xml))

(setq \_robot\_namespace (string \_\_robot\_namespace))

(setq \_initial\_pose \_\_initial\_pose)

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

self)

(:model\_name

(&optional \_\_model\_name)

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

(:model\_xml

(&optional \_\_model\_xml)

(if \_\_model\_xml (setq \_model\_xml \_\_model\_xml)) \_model\_xml)

(:robot\_namespace

(&optional \_\_robot\_namespace)

(if \_\_robot\_namespace (setq \_robot\_namespace \_\_robot\_namespace)) \_robot\_namespace)

(:initial\_pose

(&rest \_\_initial\_pose)

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

(send\* \_initial\_pose \_\_initial\_pose)

(progn

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

\_initial\_pose)))

(:reference\_frame

(&optional \_\_reference\_frame)

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

(:serialization-length

()

(+

;; string \_model\_name

4 (length \_model\_name)

;; string \_model\_xml

4 (length \_model\_xml)

;; string \_robot\_namespace

4 (length \_robot\_namespace)

;; geometry\_msgs/Pose \_initial\_pose

(send \_initial\_pose :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)

;; string \_model\_xml

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

;; string \_robot\_namespace

(write-long (length \_robot\_namespace) s) (princ \_robot\_namespace s)

;; geometry\_msgs/Pose \_initial\_pose

(send \_initial\_pose :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))

;; string \_model\_xml

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

;; string \_robot\_namespace

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

;; geometry\_msgs/Pose \_initial\_pose

(send \_initial\_pose :deserialize buf ptr-) (incf ptr- (send \_initial\_pose :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)

)

(defclass gazebo\_msgs\_new::SpawnModelResponse

:super ros::object

:slots (\_success \_status\_message ))

(defmethod gazebo\_msgs\_new::SpawnModelResponse

(:init

(&key

((:success \_\_success) nil)

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

)

(send-super :init)

(setq \_success \_\_success)

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

self)

(: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

()

(+

;; 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)))))

;; 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))

;; 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::SpawnModel

:super ros::object

:slots ())

(setf (get gazebo\_msgs\_new::SpawnModel :md5sum-) "9ed9c82c96abe1a00c3e8cdaeee24413")

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

(setf (get gazebo\_msgs\_new::SpawnModel :request) gazebo\_msgs\_new::SpawnModelRequest)

(setf (get gazebo\_msgs\_new::SpawnModel :response) gazebo\_msgs\_new::SpawnModelResponse)

(defmethod gazebo\_msgs\_new::SpawnModelRequest

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

(setf (get gazebo\_msgs\_new::SpawnModelRequest :md5sum-) "9ed9c82c96abe1a00c3e8cdaeee24413")

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

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

"string model\_name # name of the model to be spawn

string model\_xml # this should be an urdf or gazebo xml

string robot\_namespace # spawn robot and all ROS interfaces under this namespace

geometry\_msgs/Pose initial\_pose # only applied to canonical body

string reference\_frame # initial\_pose is defined relative to the frame of this model/body

# if left empty or \"world\", then gazebo world frame is used

# if non-existent model/body is specified, an error is returned

# and the model is not spawned

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

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

---

bool success # return true if spawn successful

string status\_message # comments if available

")

(setf (get gazebo\_msgs\_new::SpawnModelResponse :md5sum-) "9ed9c82c96abe1a00c3e8cdaeee24413")

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

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

"string model\_name # name of the model to be spawn

string model\_xml # this should be an urdf or gazebo xml

string robot\_namespace # spawn robot and all ROS interfaces under this namespace

geometry\_msgs/Pose initial\_pose # only applied to canonical body

string reference\_frame # initial\_pose is defined relative to the frame of this model/body

# if left empty or \"world\", then gazebo world frame is used

# if non-existent model/body is specified, an error is returned

# and the model is not spawned

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

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

---

bool success # return true if spawn successful

string status\_message # comments if available

")

(provide :gazebo\_msgs\_new/SpawnModel "9ed9c82c96abe1a00c3e8cdaeee24413")