;; Auto-generated. Do not edit!

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

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

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

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

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

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

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

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

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

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

(in-package "ROS")

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

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

(defclass gazebo\_msgs\_new::SetPhysicsPropertiesRequest

:super ros::object

:slots (\_time\_step \_max\_update\_rate \_gravity \_ode\_config ))

(defmethod gazebo\_msgs\_new::SetPhysicsPropertiesRequest

(:init

(&key

((:time\_step \_\_time\_step) 0.0)

((:max\_update\_rate \_\_max\_update\_rate) 0.0)

((:gravity \_\_gravity) (instance geometry\_msgs::Vector3 :init))

((:ode\_config \_\_ode\_config) (instance gazebo\_msgs\_new::ODEPhysics :init))

)

(send-super :init)

(setq \_time\_step (float \_\_time\_step))

(setq \_max\_update\_rate (float \_\_max\_update\_rate))

(setq \_gravity \_\_gravity)

(setq \_ode\_config \_\_ode\_config)

self)

(:time\_step

(&optional \_\_time\_step)

(if \_\_time\_step (setq \_time\_step \_\_time\_step)) \_time\_step)

(:max\_update\_rate

(&optional \_\_max\_update\_rate)

(if \_\_max\_update\_rate (setq \_max\_update\_rate \_\_max\_update\_rate)) \_max\_update\_rate)

(:gravity

(&rest \_\_gravity)

(if (keywordp (car \_\_gravity))

(send\* \_gravity \_\_gravity)

(progn

(if \_\_gravity (setq \_gravity (car \_\_gravity)))

\_gravity)))

(:ode\_config

(&rest \_\_ode\_config)

(if (keywordp (car \_\_ode\_config))

(send\* \_ode\_config \_\_ode\_config)

(progn

(if \_\_ode\_config (setq \_ode\_config (car \_\_ode\_config)))

\_ode\_config)))

(:serialization-length

()

(+

;; float64 \_time\_step

8

;; float64 \_max\_update\_rate

8

;; geometry\_msgs/Vector3 \_gravity

(send \_gravity :serialization-length)

;; gazebo\_msgs\_new/ODEPhysics \_ode\_config

(send \_ode\_config :serialization-length)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; float64 \_time\_step

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

;; float64 \_max\_update\_rate

(sys::poke \_max\_update\_rate (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;; geometry\_msgs/Vector3 \_gravity

(send \_gravity :serialize s)

;; gazebo\_msgs\_new/ODEPhysics \_ode\_config

(send \_ode\_config :serialize s)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; float64 \_time\_step

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

;; float64 \_max\_update\_rate

(setq \_max\_update\_rate (sys::peek buf ptr- :double)) (incf ptr- 8)

;; geometry\_msgs/Vector3 \_gravity

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

;; gazebo\_msgs\_new/ODEPhysics \_ode\_config

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

;;

self)

)

(defclass gazebo\_msgs\_new::SetPhysicsPropertiesResponse

:super ros::object

:slots (\_success \_status\_message ))

(defmethod gazebo\_msgs\_new::SetPhysicsPropertiesResponse

(: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::SetPhysicsProperties

:super ros::object

:slots ())

(setf (get gazebo\_msgs\_new::SetPhysicsProperties :md5sum-) "97e2057080558ce4730434b5fae75c91")

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

(setf (get gazebo\_msgs\_new::SetPhysicsProperties :request) gazebo\_msgs\_new::SetPhysicsPropertiesRequest)

(setf (get gazebo\_msgs\_new::SetPhysicsProperties :response) gazebo\_msgs\_new::SetPhysicsPropertiesResponse)

(defmethod gazebo\_msgs\_new::SetPhysicsPropertiesRequest

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

(setf (get gazebo\_msgs\_new::SetPhysicsPropertiesRequest :md5sum-) "97e2057080558ce4730434b5fae75c91")

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

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

"# sets pose and twist of a link. All children link poses/twists of the URDF tree will be updated accordingly

float64 time\_step # dt in seconds

float64 max\_update\_rate # throttle maximum physics update rate

geometry\_msgs/Vector3 gravity # gravity vector (e.g. earth ~[0,0,-9.81])

gazebo\_msgs\_new/ODEPhysics ode\_config # configurations for ODE

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

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

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

MSG: gazebo\_msgs\_new/ODEPhysics

bool auto\_disable\_bodies # enable auto disabling of bodies, default false

uint32 sor\_pgs\_precon\_iters # preconditioning inner iterations when uisng projected Gauss Seidel

uint32 sor\_pgs\_iters # inner iterations when uisng projected Gauss Seidel

float64 sor\_pgs\_w # relaxation parameter when using projected Gauss Seidel, 1 = no relaxation

float64 sor\_pgs\_rms\_error\_tol # rms error tolerance before stopping inner iterations

float64 contact\_surface\_layer # contact \"dead-band\" width

float64 contact\_max\_correcting\_vel # contact maximum correction velocity

float64 cfm # global constraint force mixing

float64 erp # global error reduction parameter

uint32 max\_contacts # maximum contact joints between two geoms

---

bool success # return true if set wrench successful

string status\_message # comments if available

")

(setf (get gazebo\_msgs\_new::SetPhysicsPropertiesResponse :md5sum-) "97e2057080558ce4730434b5fae75c91")

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

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

"# sets pose and twist of a link. All children link poses/twists of the URDF tree will be updated accordingly

float64 time\_step # dt in seconds

float64 max\_update\_rate # throttle maximum physics update rate

geometry\_msgs/Vector3 gravity # gravity vector (e.g. earth ~[0,0,-9.81])

gazebo\_msgs\_new/ODEPhysics ode\_config # configurations for ODE

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

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

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

MSG: gazebo\_msgs\_new/ODEPhysics

bool auto\_disable\_bodies # enable auto disabling of bodies, default false

uint32 sor\_pgs\_precon\_iters # preconditioning inner iterations when uisng projected Gauss Seidel

uint32 sor\_pgs\_iters # inner iterations when uisng projected Gauss Seidel

float64 sor\_pgs\_w # relaxation parameter when using projected Gauss Seidel, 1 = no relaxation

float64 sor\_pgs\_rms\_error\_tol # rms error tolerance before stopping inner iterations

float64 contact\_surface\_layer # contact \"dead-band\" width

float64 contact\_max\_correcting\_vel # contact maximum correction velocity

float64 cfm # global constraint force mixing

float64 erp # global error reduction parameter

uint32 max\_contacts # maximum contact joints between two geoms

---

bool success # return true if set wrench successful

string status\_message # comments if available

")

(provide :gazebo\_msgs\_new/SetPhysicsProperties "97e2057080558ce4730434b5fae75c91")