;; Auto-generated. Do not edit!

(when (boundp 'ur\_msgs\_new::SetSpeedSliderFraction)

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

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

(shadow 'SetSpeedSliderFraction (find-package "UR\_MSGS\_NEW")))

(unless (find-package "UR\_MSGS\_NEW::SETSPEEDSLIDERFRACTION")

(make-package "UR\_MSGS\_NEW::SETSPEEDSLIDERFRACTION"))

(unless (find-package "UR\_MSGS\_NEW::SETSPEEDSLIDERFRACTIONREQUEST")

(make-package "UR\_MSGS\_NEW::SETSPEEDSLIDERFRACTIONREQUEST"))

(unless (find-package "UR\_MSGS\_NEW::SETSPEEDSLIDERFRACTIONRESPONSE")

(make-package "UR\_MSGS\_NEW::SETSPEEDSLIDERFRACTIONRESPONSE"))

(in-package "ROS")

(defclass ur\_msgs\_new::SetSpeedSliderFractionRequest

:super ros::object

:slots (\_speed\_slider\_fraction ))

(defmethod ur\_msgs\_new::SetSpeedSliderFractionRequest

(:init

(&key

((:speed\_slider\_fraction \_\_speed\_slider\_fraction) 0.0)

)

(send-super :init)

(setq \_speed\_slider\_fraction (float \_\_speed\_slider\_fraction))

self)

(:speed\_slider\_fraction

(&optional \_\_speed\_slider\_fraction)

(if \_\_speed\_slider\_fraction (setq \_speed\_slider\_fraction \_\_speed\_slider\_fraction)) \_speed\_slider\_fraction)

(:serialization-length

()

(+

;; float64 \_speed\_slider\_fraction

8

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; float64 \_speed\_slider\_fraction

(sys::poke \_speed\_slider\_fraction (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; float64 \_speed\_slider\_fraction

(setq \_speed\_slider\_fraction (sys::peek buf ptr- :double)) (incf ptr- 8)

;;

self)

)

(defclass ur\_msgs\_new::SetSpeedSliderFractionResponse

:super ros::object

:slots (\_success ))

(defmethod ur\_msgs\_new::SetSpeedSliderFractionResponse

(:init

(&key

((:success \_\_success) nil)

)

(send-super :init)

(setq \_success \_\_success)

self)

(:success

(&optional \_\_success)

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

(:serialization-length

()

(+

;; bool \_success

1

))

(: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))

;;

(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)

;;

self)

)

(defclass ur\_msgs\_new::SetSpeedSliderFraction

:super ros::object

:slots ())

(setf (get ur\_msgs\_new::SetSpeedSliderFraction :md5sum-) "172aeb6c49379a44cf68480fa5bfad3c")

(setf (get ur\_msgs\_new::SetSpeedSliderFraction :datatype-) "ur\_msgs\_new/SetSpeedSliderFraction")

(setf (get ur\_msgs\_new::SetSpeedSliderFraction :request) ur\_msgs\_new::SetSpeedSliderFractionRequest)

(setf (get ur\_msgs\_new::SetSpeedSliderFraction :response) ur\_msgs\_new::SetSpeedSliderFractionResponse)

(defmethod ur\_msgs\_new::SetSpeedSliderFractionRequest

(:response () (instance ur\_msgs\_new::SetSpeedSliderFractionResponse :init)))

(setf (get ur\_msgs\_new::SetSpeedSliderFractionRequest :md5sum-) "172aeb6c49379a44cf68480fa5bfad3c")

(setf (get ur\_msgs\_new::SetSpeedSliderFractionRequest :datatype-) "ur\_msgs\_new/SetSpeedSliderFractionRequest")

(setf (get ur\_msgs\_new::SetSpeedSliderFractionRequest :definition-)

"# Set the speed slider on the teach pendant to the specified value.

#

# Values for 'speed\_slider\_fraction' must be from [0; 1.0]; values outside this

# valid range will result in an error being returned to the caller.

float64 speed\_slider\_fraction

---

bool success

")

(setf (get ur\_msgs\_new::SetSpeedSliderFractionResponse :md5sum-) "172aeb6c49379a44cf68480fa5bfad3c")

(setf (get ur\_msgs\_new::SetSpeedSliderFractionResponse :datatype-) "ur\_msgs\_new/SetSpeedSliderFractionResponse")

(setf (get ur\_msgs\_new::SetSpeedSliderFractionResponse :definition-)

"# Set the speed slider on the teach pendant to the specified value.

#

# Values for 'speed\_slider\_fraction' must be from [0; 1.0]; values outside this

# valid range will result in an error being returned to the caller.

float64 speed\_slider\_fraction

---

bool success

")

(provide :ur\_msgs\_new/SetSpeedSliderFraction "172aeb6c49379a44cf68480fa5bfad3c")