;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::named\_input\_output)

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

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

(shadow 'named\_input\_output (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::NAMED\_INPUT\_OUTPUT")

(make-package "ROBOTNIK\_MSGS\_NEW::NAMED\_INPUT\_OUTPUT"))

(in-package "ROS")

;;//! \htmlinclude named\_input\_output.msg.html

(defclass robotnik\_msgs\_new::named\_input\_output

:super ros::object

:slots (\_name \_value ))

(defmethod robotnik\_msgs\_new::named\_input\_output

(:init

(&key

((:name \_\_name) "")

((:value \_\_value) nil)

)

(send-super :init)

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

(setq \_value \_\_value)

self)

(:name

(&optional \_\_name)

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

(:value

(&optional \_\_value)

(if \_\_value (setq \_value \_\_value)) \_value)

(:serialization-length

()

(+

;; string \_name

4 (length \_name)

;; bool \_value

1

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; string \_name

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

;; bool \_value

(if \_value (write-byte -1 s) (write-byte 0 s))

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; string \_name

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

;; bool \_value

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

;;

self)

)

(setf (get robotnik\_msgs\_new::named\_input\_output :md5sum-) "23f05028c1a699fb83e22401228c3a9e")

(setf (get robotnik\_msgs\_new::named\_input\_output :datatype-) "robotnik\_msgs\_new/named\_input\_output")

(setf (get robotnik\_msgs\_new::named\_input\_output :definition-)

"string name

bool value

")

(provide :robotnik\_msgs\_new/named\_input\_output "23f05028c1a699fb83e22401228c3a9e")