;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::StringArray)

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

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

(shadow 'StringArray (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::STRINGARRAY")

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

(in-package "ROS")

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

(defclass robotnik\_msgs\_new::StringArray

:super ros::object

:slots (\_text ))

(defmethod robotnik\_msgs\_new::StringArray

(:init

(&key

((:text \_\_text) (let (r) (dotimes (i 0) (push "" r)) r))

)

(send-super :init)

(setq \_text \_\_text)

self)

(:text

(&optional \_\_text)

(if \_\_text (setq \_text \_\_text)) \_text)

(:serialization-length

()

(+

;; string[] \_text

(apply #'+ (mapcar #'(lambda (x) (+ 4 (length x))) \_text)) 4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; string[] \_text

(write-long (length \_text) s)

(dolist (elem \_text)

(write-long (length elem) s) (princ elem s)

)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; string[] \_text

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_text (make-list n))

(dotimes (i n)

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setf (elt \_text i) (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

))

;;

self)

)

(setf (get robotnik\_msgs\_new::StringArray :md5sum-) "da62ad767a4dbff3a3ce44939e6c55e8")

(setf (get robotnik\_msgs\_new::StringArray :datatype-) "robotnik\_msgs\_new/StringArray")

(setf (get robotnik\_msgs\_new::StringArray :definition-)

"string[] text

")

(provide :robotnik\_msgs\_new/StringArray "da62ad767a4dbff3a3ce44939e6c55e8")