;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::Pose2DArray)

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

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

(shadow 'Pose2DArray (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::POSE2DARRAY")

(make-package "ROBOTNIK\_MSGS\_NEW::POSE2DARRAY"))

(in-package "ROS")

;;//! \htmlinclude Pose2DArray.msg.html

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

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

(defclass robotnik\_msgs\_new::Pose2DArray

:super ros::object

:slots (\_poses ))

(defmethod robotnik\_msgs\_new::Pose2DArray

(:init

(&key

((:poses \_\_poses) (let (r) (dotimes (i 0) (push (instance geometry\_msgs::Pose2D :init) r)) r))

)

(send-super :init)

(setq \_poses \_\_poses)

self)

(:poses

(&rest \_\_poses)

(if (keywordp (car \_\_poses))

(send\* \_poses \_\_poses)

(progn

(if \_\_poses (setq \_poses (car \_\_poses)))

\_poses)))

(:serialization-length

()

(+

;; geometry\_msgs/Pose2D[] \_poses

(apply #'+ (send-all \_poses :serialization-length)) 4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

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

;; geometry\_msgs/Pose2D[] \_poses

(write-long (length \_poses) s)

(dolist (elem \_poses)

(send elem :serialize s)

)

;;

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

(:deserialize

(buf &optional (ptr- 0))

;; geometry\_msgs/Pose2D[] \_poses

(let (n)

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

(setq \_poses (let (r) (dotimes (i n) (push (instance geometry\_msgs::Pose2D :init) r)) r))

(dolist (elem- \_poses)

(send elem- :deserialize buf ptr-) (incf ptr- (send elem- :serialization-length))

))

;;

self)

)

(setf (get robotnik\_msgs\_new::Pose2DArray :md5sum-) "31a710d661541413a823a4591402c488")

(setf (get robotnik\_msgs\_new::Pose2DArray :datatype-) "robotnik\_msgs\_new/Pose2DArray")

(setf (get robotnik\_msgs\_new::Pose2DArray :definition-)

"geometry\_msgs/Pose2D[] poses

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

MSG: geometry\_msgs/Pose2D

# Deprecated

# Please use the full 3D pose.

# In general our recommendation is to use a full 3D representation of everything and for 2D specific applications make the appropriate projections into the plane for their calculations but optimally will preserve the 3D information during processing.

# If we have parallel copies of 2D datatypes every UI and other pipeline will end up needing to have dual interfaces to plot everything. And you will end up with not being able to use 3D tools for 2D use cases even if they're completely valid, as you'd have to reimplement it with different inputs and outputs. It's not particularly hard to plot the 2D pose or compute the yaw error for the Pose message and there are already tools and libraries that can do this for you.

# This expresses a position and orientation on a 2D manifold.

float64 x

float64 y

float64 theta

")

(provide :robotnik\_msgs\_new/Pose2DArray "31a710d661541413a823a4591402c488")