Conversion Namespace Reference

The namespace **Conversion** provides some utilities to convert from one representation into another. Representations include tf, Eigen::Affine3d, std::vector, geometry_msgs::Pose, geometry_msgs::Point, and CRCL. More...

Functions

template<typename From , typename To >	
To	**Convert** (From f) Empty conversion of type from into type to. If called, asserts. More...
template<>	
tf::Pose	**Convert< Eigen::Affine3d, tf::Pose >** (Eigen::Affine3d pose) Convert Eigen::Affine3d into tf::Pose. More...
template<>	
tf::Pose	**Convert< geometry_msgs::Pose, tf::Pose >** (geometry_msgs::Pose m) Convert geometry_msgs::Pose into tf::Pose. More...
template<>	
tf::Quaternion	**Convert< Eigen::Quaterniond, tf::Quaternion >** (Eigen::Quaterniond e) Convert Eigen::Quaterniond into tf::Quaternion. More...
template<>	
tf::Pose	**Convert< tf::Quaternion, tf::Pose >** (tf::Quaternion q) Convert tf::Quaternion into tf::Pose. More...
template<>	
tf::Pose	**Convert< tf::Vector3, tf::Pose >** (tf::Vector3 t) Convert tf::Vector3 into tf::Pose. More...
tf::Pose	**CreateRPYPose** (std::vector< double > ds) CreateRPYPose taks array of double and create a tf::Pose. More...
tf::Pose	**CreatePose** (tf::Vector3 axis, double angle) Create Pose from a axis and angle rotation representation. More...
tf::Quaternion	**RPYRadians** (double roll, double pitch, double yaw)

	Create Quaternion from a rpy rotation representation designated in radians. More...
tf::Quaternion	**RPYDegrees** (double roll, double pitch, double yaw) Create Quaternion from a rpy rotation representation designated in degrees. More...
template<> tf::Vector3	**Convert< Eigen::Vector3d, tf::Vector3 >** (Eigen::Vector3d e) Convert geometry_msgs::Pose into tf::Vector3. More...
template<> tf::Pose	**Convert< Eigen::Vector3d, tf::Pose >** (Eigen::Vector3d e) Convert Eigen::Vector3d into tf::Pose. More...
template<typename T > tf::Vector3	**matrixEigenToTfVector** (T e) Convert Eigen matrix into tf::Vector3. Example: tf::Vector3 v = matrixEigenToTfVector<Eigen::Matrix3d>(m);. More...
template<> tf::Matrix3x3	**Convert< Eigen::Matrix3d, tf::Matrix3x3 >** (Eigen::Matrix3d e) Convert Eigen Matrix3d into tf::Matrix3x3. More...
tf::Pose	**Identity** () Create Identity Pose. More...
template<> Eigen::Affine3d	**Convert< tf::Pose, Eigen::Affine3d >** (tf::Pose pose) **Convert<tf::Pose, Eigen::Affine3d>** converts tf pose into an Eigen affine 4x4 matrix o represent the pose. More...
template<> Eigen::Affine3d	**Convert< geometry_msgs::Pose, Eigen::Affine3d >** (geometry_msgs::Pose m) Convert geometry_msgs::Pose into an Eigen affine3d 4x4 matrix o represent the pose. Uses tf conversion utilities. More...
template<> Eigen::Translation3d	**Convert< geometry_msgs::Pose, Eigen::Translation3d >** (geometry_msgs::Pose pose) Convert geometry_msgs::Pose into an Eigen::Translation3d. More...
template<> Eigen::Vector3d	**Convert< tf::Vector3, Eigen::Vector3d >** (tf::Vector3 t) Convert tf::Vector3 into an Eigen::Vector3d. More...

template<>	
Eigen::Affine3d	**Convert< Eigen::Vector3d, Eigen::Affine3d >** (Eigen::Vector3d translation) Convert Eigen::Vector3d translation into an Eigen::Affine3d pose. More...
Eigen::Affine3d	**CreateEigenPose** (double zangle) Create Eigen::Affine3d as an axis angle definition around z axis. More...
template<>	
Eigen::Affine3d	**Convert< Eigen::Translation3d, Eigen::Affine3d >** (Eigen::Translation3d trans) Convert Eigen::Translation3d translation into an Eigen::Affine3d pose. More...
template<>	
Eigen::Vector3d	**Convert< geometry_msgs::Point, Eigen::Vector3d >** (geometry_msgs::Point point) Convert geometry_msgs::Point translation into an Eigen::Vector3d vector. More...
template<>	
Eigen::Affine3d	**Convert< geometry_msgs::Point, Eigen::Affine3d >** (geometry_msgs::Point point) Convert geometry_msgs::Point translation into an Eigen::Affine3d pose. More...
template<>	
geometry_msgs::Pose	**Convert< tf::Pose, geometry_msgs::Pose >** (tf::Pose m) Convert tf::Pose pose into an geometry_msgs::Pose pose. More...
template<>	
geometry_msgs::Pose	**Convert< geometry_msgs::Point, geometry_msgs::Pose >** (geometry_msgs::Point point) Convert geometry_msgs::Point point into an geometry_msgs::Pose pose. More...
template<>	
geometry_msgs::Point	**Convert< Eigen::Vector3d, geometry_msgs::Point >** (Eigen::Vector3d point) Convert Eigen::Vector3d point into an geometry_msgs::Point position vector. More...
template<>	

geometry_msgs::Pose	**Convert< Eigen::Affine3d, geometry_msgs::Pose >** (Eigen::Affine3d e) Convert Eigen::Affine3d pose into an geometry_msgs::Pose pose. More...
template<> geometry_msgs::Point	**Convert< Eigen::Affine3d, geometry_msgs::Point >** (Eigen::Affine3d pose) Convert Eigen::Affine3d pose into an geometry_msgs::Point translation element. More...
template<> JointState	**Convert< std::vector< double >, JointState >** (std::vector< double > src) Convert array of std::vector<double> doubles into an JointState position, but blanking velcity, and effort. More...

Detailed Description

The namespace **Conversion** provides some utilities to convert from one representation into another. Representations include tf, Eigen::Affine3d, std::vector, geometry_msgs::Pose, geometry_msgs::Point, and CRCL.

Clearly there may be faster const references, but they require special line to convert, cannot be done in line since you cannot pass a const reference to a constructor on the stack in g++ unless you override a warning.

For g++, compilation would be faster if these conversion routines were placed in source file (cpp) OR you used precompiled header in g++. here is a "silent" error when exceeding precompiled header limits in g++. (Or was at one time).

Function Documentation

template<typename From , typename To >

To Conversion::Convert (From f) inline

Empty conversion of type from into type to. If called, asserts.

Parameters

from is defined in the template corresponding typename.

Returns

to is defined in the template corresponding typename

template<>

geometry_msgs::Point Conversion::Convert< Eigen::Affine3d, geometry_msgs::Point > (Eigen::Affine3d pose) inline

Convert Eigen::Affine3d pose into an geometry_msgs::Point translation element.

Parameters

e is Eigen::Affine3d defining pose.

Returns

geometry_msgs::Point translation element.

template<>

geometry_msgs::Pose Conversion::Convert< Eigen::Affine3d, geometry_msgs::Pose > (Eigen::Affine3d e) inline

Convert Eigen::Affine3d pose into an geometry_msgs::Pose pose.

Parameters

e is Eigen::Affine3d defining equivalent pose.

Returns

geometry_msgs::Pose pose.

template<>
tf::Pose Conversion::Convert< Eigen::Affine3d, tf::Pose > **(Eigen::Affine3d pose)** inline

Convert Eigen::Affine3d into tf::Pose.

Parameters

pose is copy constructor of Eigen::Affine3d.

Returns

tf::Pose

template<>
tf::Matrix3x3 Conversion::Convert< Eigen::Matrix3d, tf::Matrix3x3 > **(Eigen::Matrix3d e)** inline

Convert Eigen Matrix3d into tf::Matrix3x3.

Parameters

e is copy constructor of Eigen Matrix3d, a 3x3 double matrix.

Returns

tf::Matrix3x3

template<>
tf::Quaternion Conversion::Convert< Eigen::Quaterniond, tf::Quaternion > **(Eigen::Quaterniond e)** inline

Convert Eigen::Quaterniond into tf::Quaternion.

Parameters

e is copy constructor of Eigen::Quaterniond.

Returns

tf::Quaternion

template<>

Eigen::Affine3d Conversion::Convert< Eigen::Translation3d, Eigen::Affine3d > (Eigen::Translation3d trans) inline

Convert Eigen::Translation3d translation into an Eigen::Affine3d pose.

Parameters

t is translation defined as a Eigen::Translation3d.

Returns

Eigen::Affine3d pose

template<>

Eigen::Affine3d Conversion::Convert< Eigen::Vector3d, Eigen::Affine3d > (Eigen::Vector3d translation) inline

Convert Eigen::Vector3d translation into an Eigen::Affine3d pose.

Parameters

translation is defined as a Eigen::Vector3d.

Returns

Eigen::Affine3d pose

template<>

geometry_msgs::Point Conversion::Convert< Eigen::Vector3d, geometry_msgs::Point > (Eigen::Vector3d point) inline

Convert Eigen::Vector3d point into an geometry_msgs::Point position vector.

Parameters

point Eigen::Vector3d is translation.

Returns

geometry_msgs::Point position vector.

template<>

tf::Pose Conversion::Convert< Eigen::Vector3d, tf::Pose > (Eigen::Vector3d e) inline

Convert Eigen::Vector3d into tf::Pose.

Parameters

e is copy constructor of Eigen::Vector3d.

Returns

tf::Pose

template<>

tf::Vector3 Conversion::Convert< Eigen::Vector3d, tf::Vector3 > (Eigen::Vector3d e) inline

Convert geometry_msgs::Pose into tf::Vector3.

Parameters

e is copy constructor of Eigen::Vector3d.

Returns

tf::Vector3

template<>

Eigen::Affine3d Conversion::Convert< geometry_msgs::Point, Eigen::Affine3d > (geometry_msgs::Point point) inline

Convert geometry_msgs::Point translation into an Eigen::Affine3d pose.

Parameters

point is translation defined as a geometry_msgs::Point.

Returns

Eigen::Affine3d pose

template<>

Eigen::Vector3d Conversion::Convert< geometry_msgs::Point, Eigen::Vector3d > (geometry_msgs::Point point) inline

Convert geometry_msgs::Point translation into an Eigen::Vector3d vector.

Parameters

point is translation defined as a geometry_msgs::Point.

Returns

Eigen::Vector3d vector

template<>

geometry_msgs::Pose Conversion::Convert< geometry_msgs::Point, geometry_msgs::Pose > (geometry_msgs::Point point) inline

Convert geometry_msgs::Point point into an geometry_msgs::Pose pose.

Parameters

point geometry_msgs::Point is translation.

Returns

geometry_msgs::Pose pose.

template<>

Eigen::Affine3d Conversion::Convert< geometry_msgs::Pose, Eigen::Affine3d > (geometry_msgs::Pose m) inline

Convert geometry_msgs::Pose into an Eigen affine3d 4x4 matrix o represent the pose. Uses tf conversion utilities.

Parameters

m is defined as a geometry_msgs::Pose..

Returns

Eigen Affine3d pose

template<>
Eigen::Translation3d Conversion::Convert< geometry_msgs::Pose, Eigen::Translation3d > (geometry_msgs::Pose pose) inline

Convert geometry_msgs::Pose into an Eigen::Translation3d.

Parameters

pose is defined as a geometry_msgs::Pose..

Returns

Eigen::Translation3d vector

template<>
tf::Pose Conversion::Convert< geometry_msgs::Pose, tf::Pose > (geometry_msgs::Pose m) inline

Convert geometry_msgs::Pose into tf::Pose.

Parameters

pose is copy constructor of geometry_msgs::Pose.

Returns

tf::Pose

template<>
JointState Conversion::Convert< std::vector< double >, JointState > (std::vector< double > src) inline

Convert array of std::vector<double> doubles into an JointState position, but blanking velcity, and effort.

Parameters

src is a std::vector of doubles defining the value for each joint.

Returns

sensor_msgs::JointState_<std::allocator<void> > definition.

template<>
Eigen::Affine3d Conversion::Convert< tf::Pose, Eigen::Affine3d > (tf::Pose pose) inline

Convert<tf::Pose, Eigen::Affine3d> converts tf pose into an Eigen affine 4x4 matrix o represent the pose.

Parameters

pose is the tf pose with position and orientation.

Returns

Eigen Affine3d pose

template<>
geometry_msgs::Pose Conversion::Convert< tf::Pose, geometry_msgs::Pose > (tf::Pose m) inline

Convert tf::Pose pose into an geometry_msgs::Pose pose.

Parameters

m is a tf::Pose transform matrix.

Returns

geometry_msgs::Pose pose

template<>
tf::Pose Conversion::Convert< tf::Quaternion, tf::Pose > (tf::Quaternion q) inline

Convert tf::Quaternion into tf::Pose.

Parameters

q rotation is converted into a tf::Pose.

Returns

tf::Pose

template<>
Eigen::Vector3d Conversion::Convert< tf::Vector3, Eigen::Vector3d > (tf::Vector3 t) inline

Convert tf::Vector3 into an Eigen::Vector3d.

Parameters

t is translation is defined as a tf::Vector3..

Returns

Eigen::Vector3d vector

template<>
tf::Pose Conversion::Convert< tf::Vector3, tf::Pose > (tf::Vector3 t) inline

Convert tf::Vector3 into tf::Pose.

Parameters

t is translation is converted into a tf::Pose.

Returns

tf::Pose

Eigen::Affine3d Conversion::CreateEigenPose (double zangle) inline

Create Eigen::Affine3d as an axis angle definition around z axis.

Parameters

zangle is angle of rotation in radians around Z.

Returns

Eigen::Affine3d pose

tf::Pose Conversion::CreatePose (tf::Vector3 axis, double angle) inline

Create Pose from a axis and angle rotation representation.

Parameters

axis	is the unit vector to rotation around.
angle	is the angle of rotation in radians.

Returns

tf::Pose

tf::Pose Conversion::CreateRPYPose (std::vector< double > ds) inline

CreateRPYPose taks array of double and create a tf::Pose.

Parameters

ds is a std array of 6 doubles to create pose (rpy + xyz).

Returns

tf::Pose

tf::Pose Conversion::Identity ( ) inline

Create Identity Pose.

Returns

tf::Pose

template<typename T >

tf::Vector3 Conversion::matrixEigenToTfVector (T e) inline

Convert Eigen matrix into tf::Vector3. Example: tf::Vector3 v = matrixEigenToTfVector<Eigen::Matrix3d>(m);.

Parameters

e	is copy constructor of Eigen Matrix, either 3x3, 4x4, double or float.

Returns

tf::Vector3

tf::Quaternion Conversion::RPYDegrees (double roll, double pitch, double yaw) inline

Create Quaternion from a rpy rotation representation designated in degrees.

Parameters

roll	rotation around x axis in degrees.
pitch	rotation around y axis in degrees.
yaw	rotation around z axis in degrees.

Returns

tf::Quaternion

tf::Quaternion Conversion::RPYRadians (double roll,
double pitch,
double yaw
) inline

Create Quaternion from a rpy rotation representation designated in radians.

Parameters

roll	rotation around x axis in radians.
pitch	rotation around y axis in radians.
yaw	rotation around z axis in radians.

Returns

tf::Quaternion