

# ROSSerial provider

## ROS(Robot Operating System)

Version 1.0.0

User's guide

September 8, 2017

[Remarks]



**[Revision history]**

Version	Date	Contents
1.0.0	2017-09-08	First edition

**[Supported model]**

Model	Version	Remarks

---

## Contents

1. Introduction .....	4
2. Overview of ROSSerial provider .....	5
2.1. Overview .....	5
2.2. Methods and Properties.....	6
2.2.1. CaoWorkspace::AddController method .....	6
2.2.2. CaoController::AddVariable method .....	6
2.2.3. CaoController::AddExtension method.....	7
2.2.4. CaoExtension::AddVariable method.....	8
2.2.5. CaoVariable::get_Value property .....	8
2.2.6. CaoVariable::put_Value property .....	9
Appendix A. Sample program.....	9
Appendix B. Open source copyright information .....	10

## 1. Introduction

This document is the user's guide of the CAO provider that communicates with ROS (Robot Operating System). The CAO provider (CaoProvROSSerial.dll) handled in this document is called the ROSSerial provider.

The next chapter contains an overview of the ROSSerial provider.

## 2. Overview of ROSSerial provider

### 2.1. Overview

The ROSSerial provider is a CAO provider that handles ROS messages in conjunction with `rosserial_server` on Linux. Its file format is a Dynamic Link Library (DLL), which is dynamically loaded from the CAO engine when it is used. To use the ROSSerial provider, you need to install ORiN 2 SDK or manually register registry by referring to the table below.

**Table 2-1 ROSSerial provider**

File name	CaoProvROSSerial.dll
ProgID	CaoProv.ROS.ROSSerial
Registration	regsvr32 CaoProvROSSerial.dll
Deregistration	regsvr32 /u CaoProvROSSerial.dll



Mode=<Action mode>	Required. Specifies the action mode of the generated topic. (1: Publisher, 2: Subscriber)
Type=<Data type>	Required. Specify the data type of the generated topic. Table 2-4 shows the correspondence between data types and ROS messages.

**Table 2-4 Correspondence between data type and ROS message**

Data type	ROS message
2 (VT_I2)	std_msgs/Int16
3 (VT_I4)	std_msgs/Int32
4 (VT_R4)	std_msgs/Float32
5 (VT_R8)	std_msgs/Float64
8 (VT_BSTR)	std_msgs/String
11 (VT_BOOL)	std_msgs/Bool
16 (VT_I1)	std_msgs/Int8
17 (VT_UI1)	std_msgs/UInt8
18 (VT_UI2)	std_msgs/UInt16
19 (VT_UI4)	std_msgs/UInt32
20 (VT_I8)	std_msgs/Int64
21 (VT_UI8)	std_msgs/UInt64
8194 (VT_I2   VT_ARRAY)	std_msgs/Int16MultiArray
8195 (VT_I4   VT_ARRAY)	std_msgs/Int32MultiArray
8196 (VT_R4   VT_ARRAY)	std_msgs/Float32MultiArray
8197 (VT_R8   VT_ARRAY)	std_msgs/Float64MultiArray
8208 (VT_I1   VT_ARRAY)	std_msgs/Int8MultiArray
8209 (VT_UI1   VT_ARRAY)	std_msgs/UInt8MultiArray
8210 (VT_UI2   VT_ARRAY)	std_msgs/UInt16MultiArray
8211 (VT_UI4   VT_ARRAY)	std_msgs/UInt32MultiArray
8212 (VT_I8   VT_ARRAY)	std_msgs/Int64MultiArray
8213 (VT_UI8   VT_ARRAY)	std_msgs/UInt64MultiArray

### 2.2.3. CaoController::AddExtension method

The argument of the AddExtension method of the CaoController class specifies the ROS message name (BSTR type) and option string (BSTR type).

**Syntax** AddExtension( <bstrName:BSTR> [,<bstrOption:BSTR>] )

bstrName : [in] ROS message name

bstrOption : [in] Option string

The following table shows the list of option strings

**Table 2-5 CaoController::AddExtension option strings**

Option	Description
Path=<ROS root>	Specify the path of the directory containing the ROS package group. When omitted, the following values are used. "\$(ORiN2)¥CAO¥ProviderLib¥ROS¥ROSSerial¥Bin¥ros_lib"

**Example**

---

```
Dim caoExt as CaoExtension
Set caoExt = caoCtrl.AddExtension("std_msgs/ColorRGBA") ' Generate std_msgs / ColorRGBA message
```

---

#### 2.2.4. CaoExtension::AddVariable method

The AddVariable method provides the function of generating and retrieving CaoVariable objects. The argument of the AddVariable method of the CaoExtension class specifies the variable name (BSTR type) and option string (BSTR type). The "variable name" specified here is used as the topic name on ROS. The data type of the generated topic is the ROS message of CaoExtension.

**Syntax** AddVariable( <bstrName:BSTR> [,<bstrOption:BSTR>] )

bstrName : [in] Variable name

bstrOption : [in] Option string

The following table shows the list of option strings

**Table 2-6 CaoExtension::AddVariable option strings**

Option	Description
Mode=<Action mode >	Required. Specifies the action mode of the generated topic. (1: Publisher, 2: Subscriber)

#### 2.2.5. CaoVariable::get\_Value property

Retrieve the message received from the ROS. It is effective only when the operation mode is Subscriber.

### 2.2.6. CaoVariable::put\_Value property

Send a message on ROS. It is valid only when the operation mode is Publisher.

## Appendix A. Sample program

The following shows a case where a sample that transmits a value on ROS using ROSSerial provider is written in VB 6. Before running the sample you need to start `rosserial_server` on ROS. For the sample, the IP address of the PC running `rosserial_server` is set as below.

IP: 192.168.0.1

Sample	ROSSerial.frm
--------	---------------

```
Dim g_eng As CaoEngine
Dim g_ctrl As CaoController
Dim g_val As CaoVariable

Private Sub Command1_Click()
    ' Send messege
    g_val.Value = 1
End Sub

Private Sub Form_Load()
    Set g_eng = New CaoEngine

    ' Connection processing IP should be set for each controller
    Set g_ctrl = g_eng.Workspaces(0).AddController("", "CaoProv.ROS.ROSSerial", "", "Server=192.168.204.137")

    ' Publisher"/test"
    Set g_val = g_ctrl.AddVariable("/test", "Mode=1,Type=3")

End Sub

Private Sub Form_Unload(Cancel As Integer)
    ' Discard variable object
    g_ctrl.Variables.Clear
    Set g_val = Nothing

    ' Discard controller object
    g_eng.Workspaces(0).Controllers.Remove g_ctrl.Index
    Set g_ctrl = Nothing

    ' Discard CaoEngine
    Set g_eng = Nothing
End Sub
```

## Appendix B. Open source copyright information

The ROSSerial provider uses open source software `rosserial_ windows`.

The copyright information is displayed below.

---

`rosserial_ windows`.

Original code by Kareem Shehata ([http://ros.org/wiki/rosserial\\_ windows](http://ros.org/wiki/rosserial_ windows))

Copyright (c) 2014, Clearpath Robotics, Inc., All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

---