DENSO
RFID VOCS Scanner UR30 provider

Version 1.0.2

User's Guide

May 17, 2022

Revision history

Version	Date	Content
1.0.0	2019-08-01	First edition
1.0.1	2019-10-23	Refactoring of source code
1.0.2	2022-05-17	Corrected to be omitted when Empty is included as an argument of StartInventory command.

[Operation Check Model]

Model	Version	Note
URCT-M31	Ver. 1.04	

Contents

1. Introduction

This guide is a user's guide to obtaining RFID tags from RFID VOCS Scanner's UR30 series. Figure 1-1 shows the overall configuration of this provider and device. Hereinafter, this provider is called the UR30 provider, and the RFID VOCS Scanner is called the scanner. The UR30 provider performs serial communication with the scanner.

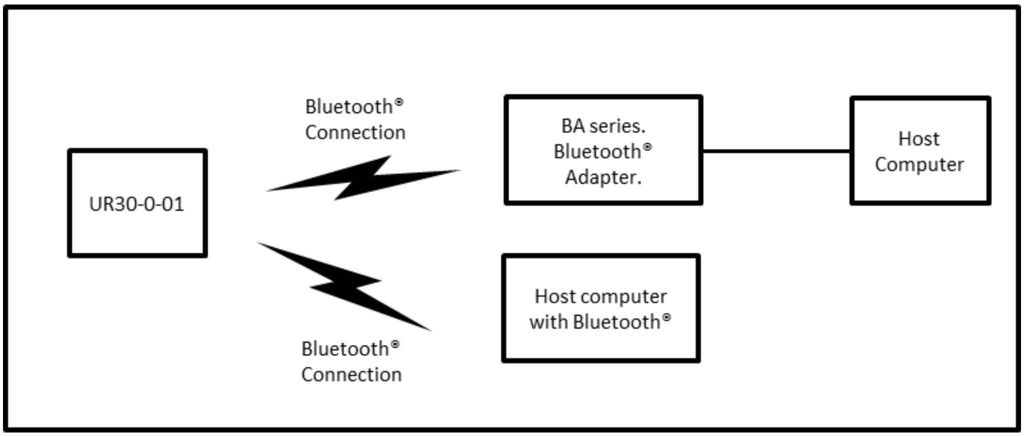

Figure 1-1 Configuration Diagram

Figure 1-2 shows the correspondence between the providers and devices.
(* This is an example. Not all are shown.)

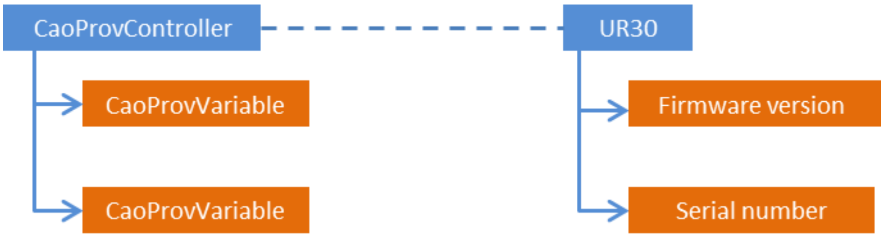

Figure 1-2 Correspondence between Provider Configuration and Device Information

1.1. Environment and version assumed by this document

It is assumed that the client PC runs on the Windows and the target CNC is a company-named device type that can be Ethernet-connected. The PC development environment can be developed if the client PC is a programming environment that supports Component Object Model (COM, Component Object Model).

1.2. Source of reference information

All examples of programming in this book are provided in Visual Basic for Applications, but can be developed in a variety of programming languages, such as C++, Java, or.NET. For instructions, see the ORiN2 Programming Guide.

The ORiN 2 Programming Guide corresponds to the following files in the ORiN 2 SDK install folder:

- ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_<lang>.pdf

※Replace the <lang> part with the environment-specific language strings.

This chapter explains the basic knowledge and techniques of ORiN2,COM/DCOM required to develop applications using providers.

2. Environment Setup for Application Development

2.1. Connecting the Scanner to the Client PC

When connecting to the scanner, start connecting to the client PC while the scanner is turned on..When using the BA series communication adapter, refer to the instruction manual for the BA series communication device.

2.2. Setting Up the PC Development Environment

2.2.1. Installing UR30 Providers Automatically

If you are in an environment with ORiN2 SDK installed, you are ready for the operating environment (Line Time) to connect to the UR30.

To set up the development environment, prepare a programming environment that supports Component Object Model (COM, Component Object Model) such as Microsoft Visual Studio 6. 0, 2003/2005/2008/2010, LabVIEW, etc.

2.2.2. Installing UR30 Providers Manually

To install UR30 providers manually, you must register the registry below. When registering with the registry, start the command prompt with administrator privileges and execute the regsvr32 command.

In addition, you must have one legitimate ORiN2 SDK license for each PC before the CAO engine can run. Refer to the "Adding or Removing a License" section in the ORiN2 SDK User's Guide.

Table 2-1 UR30 providers

File name	CaoProvDENSOUR30.dll
ProgID	CaoProv.DENSO.UR30
Registry registration	Regsvr32 CaoProvDENSOUR30.dll
Unregistering the registry	Regsvr32 /u CaoProvDENSOUR30.dll

3. Command reference

3.1. Method/Property List

Table 3-1 Method/Property List

Category	Methods/Properties[1]		Facility	Reference
CaoWorkspace				
	AddController	M	Connect to Controller	P.9
CaoController				
	VariableNames	P	Get list of variable names that can be connected	P.12
	Variables	P	Get the collection of variables held by the controller	P.12
	AddVariable	M	Adding Variable Objects	P.12
	Execute	M	Executing Extended Commands	P.13
	OnMessage	E	Message reception event	P.13
CaoVariable				
	Value	P	Get/Set Values	P.13

3.2. Method Properties

3.2.1. CaoWorkspace classes

3.2.1.1. AddController method

Add a controller object to the CaoWorkspace. UR30 providers refer to the parameters passed during the AddController method and connect to the appropriate scanner. The following are the specifications of the AddController method.

Format

CaoController AddController

```
(
        Controller name           // Controller name (optional)
        "CaoProv.DENSO.UR30",     // Provider name (fixed)
        Machine name              // Provider Execution Machine Name
        Option                    // Option string
)
```

Option

Here are the options that you specify in the option string: The option string is a string of the options shown

[1] M: Methods, P: Properties, and E: Events are shown.

below, concatenated with a comma (,).

Option	Required	Description	Values range
CONN = <Communication parameter>	✓	Specify the host information.	COM
TriggerMode=	--	Specifies the trigger mode of the access point. 0: Auto-Off 1: Momentary 2: Alternate 3: Continuous Mode 1 4: Continuous Mode 2	0-4 Default value (3)

ResponseFormat=	--	Specifies the response format of the access point.
		ResponseFormat=(a,b,c,d)
		a = PC transfer (0: Disable, 1: Enable)
		b = RSSI transmission (0: prohibited, 1: permitted)
		c = Acquired antenna information transfer (0: Disable, 1: Enable)
		d = Polarized information transfer (0: Disable, 1: Enable)
			(0,0,0,0)
			-(1,1,1,1)
			Default value (0,0,0,0)

Timaout=	--	Specifies the communication timeout in ms.	0- 4,294,967,295 Default value (1000)

Examples of uses

```
Dim engine As CaoEngine              'Engine Objects
Dim workspace As CaoWorkspace        'Workspace Objects
Dim controller As CaoController      'Controller Objects

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("SampleController",_
                                      "CaoProv.DENSO.UR30",_
                                      "",_
                                      "conn=COM:1,TriggerMode=4,ResponseFormat=(0,1,1,0)")
```

3.2.1.1.1. CONN options

The following are the connection-parameter strings for the Conn options: Here, square brackets ("[]") are optional, and the underscore in the description of each parameter indicates the default value when no option is specified.

RS232C

"Conn=COM:<COM Port>[:BaudRate>[:<Parity>:<DataBits>:<StopBits>[:Flow]]]"

<COM Port> : COM port number . '1' -COM1, '2' - COM2, ...

<BaudRate> : Communications speed 4800, 9600, 19200, 38400, 57600, 115200

<Parity> : Parity . 'N'-NONE, 'E'-EVEN, 'O'-ODD

<DataBits> : Number of the data bits '7'-7bit, '8'-8bit

<StopBits> : No. of Stop Bits . '1'-1bit, '2'-2bit

<Flow> : Flow control '0'-None,'1'-Xon/Xoff,'2'-Hardware-controlled
This can be specified by OR.

3.2.1.1.2. TriggerMode options

The following are the modes of the trigger mode.

Setting value	Name	Description
0	Auto-Off	The RF Tag communication state is activated for a maximum of 5 seconds after the trigger is pressed. The RF Tag Control is completed before 5 seconds elapse in the Communication State to enter the Communication Waiting State. Releasing the trigger during the communication state also places the device in the communication standby state.
1	Momentary	The RF Tag communication status is activated while the trigger is pressed. StartWrite,StartLock,StartKill cannot be used in this mode.
2	Alternate	Each time the trigger is pressed, the RF tag communication state and the communication standby state are alternately switched. StartWrite,StartLock,StartKill cannot be used in this mode.
3	Continuous Mode 1	The RF Tag Communication State is entered from the transmission of the Tag Function Control Start command to the transmission of the RF Tag Control Stop command. StartWrite,StartLock,StartKill cannot be used in this mode.
4	Continuous Mode 2	The RF Tag Communication State is entered from the transmission of the Tag Function Control Start command until the completion of the RF Tag control. When the RF tag control is completed, the communication standby mode is automatically set. Send the Execute command Stop(3.3.2.6) and then send the Tag Function Control Start command to enable communication again.

3.2.2. CaoController classes

3.2.2.1. VariableNames properties

Gets the list of variable names that can be connected. Variable names obtained with this property can be used as the first arguments of the AddVariable method described later.AddVariable method

Examples of uses

```
' Connection
Call Connect

' Acquisition of file name list
Dim variables as Variant
variables = controller.VariableNames

' Disconnection
Call Disconnect
```

3.2.2.2. Variables properties

Gets the collection of variables held by the controller.

Examples of uses

```
' Connection
Call Connect

' Retrieve Variable Collection
Dim variables as CaoVariables
Set variables = controller.Variables

' Get Variable
Dim variable as CaoVariable
Set variable = variables.Item(0)

' Disconnection
Call Disconnect
```

3.2.2.3. AddVariable method

Add variable objects to the CaoController. Only the variable names shown in 3.4.1 can be used.

The AddVariable specifications are as follows:

Format

```
CaoVariable AddVariable
(
        Variable name       // Variable name
        Option              // Option string (optional)
)
```

3.2.2.4. Execute method

Execute the ConController extended commands. The Execute specifications are as follows:

Format

```
Variant Execute
(
        "<extended command name>",    // Extended command name
        "<Option string>"             // Option string (optional)
)
```

Refer to 3.3 Command list for more information about the extended command list that can be specified by Execute. The examples are described in the extension command detail.

3.2.3. CaoVariable classes

3.2.3.1. Value properties

Acquires/sets data from the connected scanner. The behavior varies depending on the variable name. For details, see 3.4.Variable List.

3.2.4. OnMessage classes

The scanner issues data to be notified when an RF tag is detected during RF tag control as a message event.

For more information about events, see 3.5List of events.

Refer to 3.3.2 for the RF tag control status of the scanner.

3.3. Command list

Commands include parameter setting acquisition commands, such as parameter setting acquisition, and RF tag control commands, which enable the scanner to communicate with the RF tag.

Commanded	Description	Reference
Parameter Commands		
GetVersion	Get the firmware version. "Ver. n.nn" Nn.nn= Firmware Version	P.15
GetSirial	Get the serial number. "ID. nnnnnn" Nnnnnn = Serial number	P.16

GetSirialAndPartno	Gets the part number and serial number.
	"ID. XXXXXXXXXXnnnnnn"
	XXXXXXXXXX = Product number
	Nnnnnn = Serial number
		P.16

Commanded	Description	Reference
Parameter Commands		
GetBatteryLevel	Acquires the remaining battery level.	P.17
SetTriggerMode	Sets the trigger mode.	P.17
SetReadTypeOutput	Sets the output when reading RF tags.	P.18
GetReadTypeOutput	Retrieve the output when reading the RF tag.	P.19
SetWriteTypeOutput	Sets the RF tag write output.	P.19
GetWriteTypeOutput	Gets the output for RF tag write.	P.19
SetFrequency	Sets the center frequency. The obtained value is expressed by a bit field and sets a plurality of channels.	P.20
GetFrequency	Get the center frequency. The set value can be obtained in the bit field, and multiple channels can be set at the same time.	P.20
SetQValue	Sets the Q value used to adjust the probability of collision of responses from RF tags when communicating with RF tags.	P.21
GetQValue	Gets the Q value used to adjust the probability of collision of responses from RF tags when communicating with RF tags.	P.21
SetSession	Sets a session flag for the RF tag to determine the conditions for the re-response.	P.22
GetSession	Get the session flag that determines the condition of the re-response to the RF tag.	P.22
SetSessionInit	Sets whether to initialize the session flag.	P.23
GetSessionInit	Gets the initialization status of the session flag.	P.23
SetRereadPrevention	Sets the two-reading prevention setting for the RF tag.	P.23
GetRereadPrevention	Gets the two-reading prevention setting for the RF tag.	P.24
SetVerify	After performing "Write" on the RF tag, automatically perform "Read" and set a Verify to confirm whether writing to the RF tag was successful.	P.24
GetVerify	After "Write" is performed on the RF tag, "Read" is automatically performed to acquire Verify for confirming whether writing to the RF tag was successful.	P.25
SetPoewerSaving	Set the power saving setting for the RF reading process.	P.26
GetPoewerSaving	Gets the power-saving settings for the RF reading process.	P.26

Commanded	Description	Reference
Parameter Commands		
SetPolarization	Specify either vertical or horizontal for the antenna polarisation.	P.26
GetPolarization	Gets whether the antenna polarity is set vertically or horizontally.	P.27
SaveParameter	Saves the set parameters to the UR30 flash memory. ※The saved data is also stored in the power on/off of the UR30.	P.27
SetResponseFormat	You can change whether to acquire PC transfer, RSSI transfer, acquired antenna information transfer, or polarisation information transfer of the data to be acquired by the Tag Control Start command.	P.28
RF tag control command		
StartInventory	RF Tag Control Start Command. The UII data of the RF tag detected by the scanner is read and notified.	P.2929
StartRead	RF Tag Control Start Command. Reads and notifies the memory data specified by the RF tag detected by the scanner.	P.2929
StartWrite	RF tag control start command. Writes to the specified memory data when the scanner detects an RF tag. Returns the information of the RF tag that was written.	P.3030
StartLock	RF Tag Control Start Command. When the scanner detects an RF tag, the scanner changes the Lock status of the specified area and notifies the scanner of the tag whose Lock status has been changed.	P.3131
StartKill	RF Tag Control Start Command. Kills the tags detected when the scanner detects RF tags. Notifies you of information about killed tags. Be careful when running tags that are killed because they are no longer usable.	P.3232
Stop	When the scanner is performing RF tag control, it stops RF tag control.	P.3333
ClearBuffer	Clears the buffer for two-time read prevention.	P.3434

3.3.1. Parameter Commands

The Parameter command obtains the parameter settings for the scanner unit. If no result is returned by the scanner before the time-out period set at AddController after the command is executed, a time-out error occurs.

3.3.1.1. GetVersion

Get the firmware version of the UR30.

Item	Type Description
Argument	Without

Item	Type	Description
Returned value	VT_BSTR	Get the firmware version in "Ver.n.nn" format. n.nn = Version number

Examples of uses

```
' Connection
Call Connect
' Get firmware version
Dim version As String
version = controller.Execute("GetVersion")
' Disconnection
Call Disconnect
```

3.3.1.2. GetSerial

Gets the UR30 serial number.

Item	Type	Description
Argument	Without	
Returned value	VT_BSTR	Get the serial number in "ID. nnnnnn" format. nnnnn = Version number

Examples of uses

```
' Connection
Call Connect
' Get serial number
Dim serial As String
serial = controller.Execute("GetSerial")
' Disconnection
Call Disconnect
```

3.3.1.3. GetSerialAndPartno

Gets the UR30 part number and serial number.

Item	Type	Description
Argument	Without	

Returned value	VT_BSTR	Item number and serial number "ID. Retrieve in XXXXXXXXXXnnnnnn format
		XXXXXXXXXX = Product number
		Nnnnnn = Version number

Examples of uses

```
' Connection
Call Connect
' Get part number and serial number
Dim serialAndPart As String
serialAndPart = controller.Execute("GetSerialAndPartno")
' Disconnection
```

```
    Call Disconnect
```

3.3.1.4. GetBatteryLevel

Gets the level of the remaining battery capacity.

Item	Type	Description
Argument	Without	
Returned value	VT_I4	Acquire the remaining battery capacity. 1: Less than 10% remaining 2: From 10% to less than 40% remaining 3: Remaining 40% or more

Examples of uses

```
    ' Connection
    Call Connect
    ' Acquire Battery Remaining Level
    Dim batteryLevel As Long
    batteryLevel = controller.Execute("GetBatteryLevel")
    ' Disconnection
    Call Disconnect
```

3.3.1.5. SetTriggerMode

Sets the trigger mode for RF tag communication.

Execute before executing the RF tag control start command. If this command is executed during execution of the RF tag control start command, an error is returned.

The default is Continuous Mode 1.

Item	Type	Description
Argument	VT_I4	Specify the trigger mode. 0: Auto-Off 1: Momentary 2: Alternate 3: Continuous Mode 1 4: Continuous Mode 2 Refer to P.11 for details of each mode.11
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Set trigger mode
Call controller.Execute("SetTriggerMode", 1)
' Disconnection
Call Disconnect
```

3.3.1.6. SetReadTypeOutput

Sets the output for tag reading in units of dBm×10. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type	Description
Argument	VT_I4	Specifies the output (dBm×10) for reading. Value range: 40-240
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Set Tag Read Output
Call controller.Execute("SetReadTypeOutput", 100)
' Disconnection
Call Disconnect
```

3.3.1.7. GetReadTypeOutput

Tags read output in units of dBm×10.

Item	Type Description	
Argument	Without	
Returned value	VT_I4	Get Tag Read Output

Examples of uses

```
' Connection
Call Connect
' Get tag read output
Dim readTypeOutput As Long
readTypeOutput = controller.Execute("GetReadTypeOutput")
' Disconnection
Call Disconnect
```

3.3.1.8. SetWriteTypeOutput

Sets the output for tag writing in units of dBm×10. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description	
Argument	VT_I4	Specifies the output (dBm×10) for reading. Value range: 40-240
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Set Tag Write Output
Call controller.Execute("SetWriteTypeOutput", 100)
' Disconnection
Call Disconnect
```

3.3.1.9. GetWriteTypeOutput

Gets the output of tag write in units of dBm×10.

Item	Type Description	
Argument	Without	
Returned value	VT_I4	Get Tag Read Output

Examples of uses

```
' Connection
Call Connect
' Get tag read output
```

```
Dim writeTypeOutput As Long
writeTypeOutput = controller.Execute("GetWriteTypeOutput")
Disconnection
Call Disconnect
```

3.3.1.10. SetFrequency

Sets the center frequency (unit channel number) with the bit flag. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description	
Argument	VT_I4	Specify the center frequency to be permitted. Value Range: 0x00000001-0x0007FFFF 0x00000001 : Ch5, 0x00000002 : Ch11, 0x00000004 : Ch17, 0x00000008 : Ch23, 0x00000010 : Ch24, 0x00000020 : Ch25, 0x00000040 : Ch26, 0x00000080 : Ch27, 0x00000100 : Ch28, 0x00000200 : Ch29, 0x00000400 : Ch30, 0x00000800 : Ch31, 0x00001000 : Ch32, 0x00002000 : Ch33, 0x00004000 : Ch34, 0x00008000 : Ch35, 0x00010000 : Ch36, 0x00020000 : Ch37, 0x00040000 : Ch38
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Set Tag Write Output
' When 'Ch5,11,24 is specified
Dim frequency As Long
frequency = &H1 + &H2 + &H10
Call controller.Execute("SetFrequency", frequency)
' Disconnection
Call Disconnect
```

3.3.1.11. GetFrequency

Gets the center frequency (unit channel number) by bit flag.

Item	Type Description
Argument	Without

Item	Type Description	
Returned value	VT_I4	Get the center frequency

Examples of uses

```
' Connection
Call Connect
' Get the center frequency
Dim frequency As Long
frequency = controller.Execute("GetFrequency")
' Disconnection
Call Disconnect
```

3.3.1.12. SetQValue

Set the Q value to use to adjust the probability that the responses from the RF tags collide. A smaller value increases the collision probability, and a larger value decreases the collision probability. The higher the value, the worse the response speed may be. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description	
Argument	VT_I4	Specify a Q value. Value range: 0-7
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Set Q value
Call controller.Execute("SetQValue", 4)
' Disconnection
Call Disconnect
```

3.3.1.13. GetQValue

Gets the Q value used to adjust the probability of a response collision from an RF tag.

Item	Type Description	
Argument	Without	
Returned value	VT_I4	Get the center frequency

Examples of uses

```
' Connection
Call Connect
' Get Q value
Dim qValue As Long
qValue = controller.Execute("GetQValue")
' Disconnection
```

```
Call Disconnect
```

3.3.1.14. SetSession

Sets a session flag for the RF tag to determine the conditions for the re-response. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description	
Argument	VT_I4	Specifies the session flag. Value range: 0-3 0: Acquisition is possible again after the RF tag communication standby state. 1: The UII of the same tag can be acquired again after the prescribed time has elapsed after the RF tag is detected.[2] 2: Reacquisition is possible after a prescribed time or more after the RF tag communication standby state.[2] 3: Same as 2.
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Setting the Session Flag
Call controller.Execute("SetSession", 1)
' Disconnection
Call Disconnect
```

3.3.1.15. GetSession

Get the session flag that determines the condition of the re-response to the RF tag.

Item	Type Description	
Argument	Without	
Returned value	VT_I4	Gets the session flag.

Examples of uses

```
' Connection
Call Connect
' Get session flag
Dim session As Long
session = controller.Execute("GetSession")
' Disconnection
Call Disconnect
```

[2] The specified time is specified by the RF tag. Check the specifications of the RF tag to be used.

3.3.1.16. SetSessionInit

Set whether to initialize the session flag. By initializing the session flag each time an RF tag is detected, UII can be read continuously from the RF tag regardless of the characteristics of each session flag. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).3.3.1.26

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description	
Argument	VT_I4	Specifies whether to initialize the session flag. Value range: 0-1 0: No initialization. 1: Initialization exists.
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Set Session
Call controller.Execute("SetSessionInit", 1)
' Disconnection
Call Disconnect
```

3.3.1.17. GetSessionInit

Gets the initialization status of the session flag.

Item	Type Description	
Argument	Without	
Returned value	VT_I4	Gets the initialization status of the session flag.

Examples of uses

```
' Connection
Call Connect
' Get session flag initialization
Dim sessionInit As Long
sessionInit = controller.Execute("GetSessionInit")
' Disconnection
Call Disconnect
```

3.3.1.18. SetRereadPrevention

Changes the reading prevention setting for the RF tag twice. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description	
Argument	VT_I4	Specify the reading prevention setting twice. Value range: 0-2 0:2 read prevention setting not set 1: Continue read prevention twice during RF tag read operation. 2: Continue preventing reading twice during RF tag communication mode.
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Set to prevent reading twice
Call controller.Execute("SetRereadPrevention", 2)
' Disconnection
Call Disconnect
```

3.3.1.19. GetRereadPrevention

Acquires the twice reading prevention setting value for the RF tag.

Item	Type Description	
Argument	Without	
Returned value	VT_I4	Gets the reading prevention setting twice.

Examples of uses

```
' Connection
Call Connect
' Gets the "2-degree reading prevention setting"
Dim rereadPrevention As Integer
rereadPrevention = controller.Execute("GetRereadPrevention")
' Disconnection
Call Disconnect
```

3.3.1.20. SetVerify

Set whether to verify the RF tag during the Write process. After performing "Write" on the RF tag, automatically perform "Read" to check whether writing to the RF tag was successful. Verification takes longer than non-verification. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description

Item	Type	Description
Argument	VT_I4	Specify whether to verify. Value range: 0-1 0: No verification 1: Verified
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Verify setting
Call controller.Execute("SetVerify", 1)
' Disconnection
Call Disconnect
```

3.3.1.21. GetVerify

Acquires the twice reading prevention setting value for the RF tag.

Item	Type	Description
Argument	Without	
Returned value	VT_I4	Gets the verify status.

Examples of uses

```
' Connection
Call Connect
' Gets the "2-degree reading prevention setting"
Dim verify As Long
Verify = controller.Execute("GetVerify")
' Disconnection
Call Disconnect
```

3.3.1.22. SetPowerSaving

Specify whether to enable power-saving mode during RF read processing. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description	
Argument	VT_I4	Specify Power Save Settings. Value range: 0-1 0: Power saving disabled 1: Enable Power Save
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Power conservation settings
Call controller.Execute("SetPowerSaving", 1)
' Disconnection
Call Disconnect
```

3.3.1.23. GetPowerSaving

Get Power Save Settings.

Item	Type Description	
Argument	Without	
Returned value	VT_I4	Get Power Save Settings.

Examples of uses

```
' Connection
Call Connect
' Get Power Save Settings
Dim powerSaving As Long
PowerSaving = controller.Execute("GetPowerSaving")
' Disconnection
Call Disconnect
```

3.3.1.24. SetPolarization

Specifies which polarisation to use as the antenna polarisation. The set value is initialized when the terminal power is turned off unless it is saved in the internal flash memory (see 3.3.1.26).

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type Description

Item	Type	Description
Argument	VT_I4	Specifies the polarisation to use. Value range: 0-2 0: Vertical only 1: Horizontal only 2: Both vertical and horizontal
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Polarisation setting
Call controller.Execute("SetPolarization", 2)
' Disconnection
Call Disconnect
```

3.3.1.25. GetPolarization

Acquires the state of use of the antenna polarizations.

Item	Type	Description
Argument	Without	
Returned value	VT_I4	Acquire the used antenna polarisation

Examples of uses

```
' Connection
Call Connect
' Acquiring Antenna Polarios
Dim polarization As Long
Polarization = controller.Execute("GetPolarization")
' Disconnection
Call Disconnect
```

3.3.1.26. SaveParameter

Saves the set UR30 parameters to the flash memory. If the set value is not saved, the set value returns to the previously saved state when the power is turned off.

This command must be executed before the Tag Control Start command is executed. If this command is executed during execution of the Tag Control Start command, an error is returned.

Item	Type	Description
Argument	Without	
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Save parameter to flash memory
```

```
Call controller.Execute("SaveParameter")
' Disconnection
Call Disconnect
```

3.3.1.27. SetResponseFormat

Sets the downloaded data when OnMessage events are received. When disabled, the corresponding data will be returned in VT_EMPTY.

Item	Type Description		
Argument	VT_ARRAY | VT_VARIANT		
	1	VT_I4	Specify the transfer settings for the PC. 0: Prohibition 1: Permission
	2	VT_I4	Specify RSSI forwarding settings. 0: Prohibition 1: Permission
	3	VT_I4	Specify the transfer settings for the acquired antenna information. 0: Prohibition 1: Permission
	4	VT_I4	Specify the transfer setting of the polarity wave information. 0: Prohibition 1: Permission
Returned value	Without		

Examples of uses

```
Connection
Call Connect
'Perform transfer permission setting of return data during tag communication.
Dim responseFormat As Variant
'Allow PC and RSS transfer settings
ResponseFormat = Array(1, 1, 0, 0)

Call controller.Execute("SetResponseFormat", responseFormat)
Disconnection
Call Disconnect
```

3.3.2. RF tag control command

The RF Tag Control Command starts and stops the communication control of the scanner by executing the command. Communication control is started to enable the scanner to detect the RF tag, and when the RF tag is detected, communication with the RF tag is performed according to the control content being executed. When the scanner detects an RF tag during communication control, a message event is issued each time information is notified to notify the scanner of the information of the detected RF tag.

For details on each command, see below.

3.3.2.1. StartInventory

When the scanner detects an RF tag, the detected UII data is notified. Communication between the RF tag and the scanner is triggered according to the trigger mode setting (see 3.2.1.1.2).

When the UII data is notified from the scanner, the scanner issues a Inventory message event (see 3.5.1).

Execution of this command returns an error while RF tag control is started.

Item	Type Description	
Argument	VT_BSTR	Optional. Specifies UII data. (hexadecimal string of up to 62 bytes, set according to UII data length)
Returned value	Without	

Examples of uses

```
' Connection
Call Connect
' Start Tag Control in UII Acquisition Mode
Dim UII As String
' UII data to specify
UII = "AAAA00001234"
Call controller.Execute("StartInventory", UII)
```

3.3.2.2. StartRead

When the scanner detects an RF tag, the specified range of memory data is notified. Communication between the RF tag and the scanner is triggered according to the trigger mode setting (see 3.2.1.1.2). When the scanners notify you of the read data, issue a Read message event (see 3.5.2). Unable to execute if the scanner has started RF tag control.

Item	Type Description		
Argument	VT_ARRAY | VT_VARIANT		
	1	VT_I4	Specify Bank Value range: 0-3 0:Reserved Bank 1:UII Bank 2:TID Bank 3:User Bank
	2	VT_I4	Specify the read start position (only a multiple of 2 can be specified) Value range: 0-65535[3]
	3	VT_I4	Specify the read size (only a multiple of 2 can be specified) Value range: 2-256[3]

[3] Check the specifications of the RF tag you are using for the actual range that can be specified.

Item	Type Description		
Argument	VT_ARRAY | VT_VARIANT		
	4	VT_BSTR	Access password is specified as hexadecimal character string (authentication processing is omitted when "00000000" is specified). Value Range: "00000000"-"FFFFFFFF"

	5	VT_BOOL	Specifies whether the read data is to be received as a byte array or a string.
			TRUE = array of bytes
			FALSE = String

	6	VT_BSTR	Optional. Specifies UII data. (hexadecimal string of up to 62 bytes, set according to UII data length)
Returned value	Without		

Examples of uses

```
' Connection
Call Connect
' Start Tag Control in Read Mode
Dim readParam As Variant
' Locate memory to read
readParam = Array(0, 4, 4, "00000000", FALSE)

Call controller.Execute("StartRead", readParam)
```

3.3.2.3. StartWrite

When the scanner detects an RF tag, the specified data is written to the specified memory location..Communication between the RF tag and the scanner is triggered according to the trigger mode setting (see 3.2.1.1.2).

When writing is executed from the scanner, the UII data to which writing is executed is notified. Write message events (see 3.5.3) are issued when UII-data is notified from scanners.

Unable to execute if the scanner has started RF tag control.

Item	Type Description		
Argument	VT_ARRAY | VT_VARIANT		
	1	VT_I4	Specify Bank Value range: 0-3 0:Reserved Bank 1:UII Bank 2:TID Bank 3:User Bank
	2	VT_I4	Specify the write start position (only a multiple of 2 can be specified) Value range: 0-655353[3]
	3	VT_BSTR	Access password is specified as hexadecimal character string (authentication processing is omitted when "00000000" is specified). Value Range: "00000000"-"FFFFFFFF"
	4	VT_BSTR	Write data (specifiable only as a multiple of 2 bytes)
	5	VT_BSTR	Optional. Specifies UII data. (hexadecimal string of up to 62 bytes, set according to UII data length)
Returned value	Without		

Examples of uses

```
' Connection
Call Connect
' Start Tag Control in Write Mode
Dim writeParam As Variant
' Set access password value
writeParam = Array(0, 4, "00000000", "1234ABCD")

Call controller.Execute("StartWrite", writeParam)
```

3.3.2.4. StartLock

The Lock status of the specified target area is changed when the scanner detects an RF tag. Communication between the RF tag and the scanner is triggered according to the trigger-mode setting (see 3.2.1.1.2).

Lock message events (see 3.5.4) are issued when scanners notify you of locked UIIs. Unable to execute if the scanner has started RF tag control.

Item	Type Description		
Argument	VT_ARRAY | VT_VARIANT		
	1	VT_I4	Specify the lock target by the bit flag (multiple lock targets can be set at the same time). 0x01: Kill Password 0x02: Access-Password 0x10: UII Bank 0x20: TID Bank 0x40: User Bank
	2	VT_I4	Specify lock type 0x00: Unlock 0x01: Locks 0x10: Permanent unlock 0x11: Permanent locks
	3	VT_BSTR	Access password is specified as hexadecimal character string (authentication processing is omitted when "00000000" is specified). Value Range: "00000000"-"FFFFFFFF" (Hex)
	4	VT_BSTR	Optional. Specifies UII data. (hexadecimal string of up to 62 bytes, set according to UII data length)
Returned value	Without		

Examples of uses

```
' Connection
Call Connect
' Start Tag Control in Lock Mode
Dim lockParam As Variant
' Lock access and kill passwords
LockParam = Array(3, 1, "00000000")

Call controller.Execute("StartLock", lockParam)
```

3.3.2.5. StartKill

KILL the RF tag detected by the scanner when the scanner detects the RF tag and notify the scanner of the information of the killed RF tag. Be careful that the KILL tag cannot be read from or written to in the future.

Communication between the RF tag and the scanner is triggered according to the trigger mode setting (see 3.2.1.1.2).

When the scanners inform them of the RFID tags, Kill message events (see 3.5.5) are issued.

Unable to execute if the scanner has started RF tag control.

Item	Type Description		
Argument	VT_ARRAY | VT_VARIANT		
	1	VT_BSTR	Specifies a kill password as a hexadecimal character string (omit authentication processing when "00000000" is specified). Value Range: "00000000"-"FFFFFFFF" (Hex)
	2	VT_BSTR	Optional. Specifies UII data. (Set according to UII data length, specified in hexadecimal string of up to 62 bytes)
Returned value	Without		

Examples of uses

```
' Connection
Call Connect
' Start RF Tag Control in Kill Mode
Dim killParam As Variant
' Enter a Kill Password
KillParam = Array("1234ABCD")

Call controller.Execute("StartKill", killParam)
```

3.3.2.6. Stop

Ends the RF tag control operation.

Item	Type Description
Argument	Without
Returned value	Without

Examples of uses

```
' Connection
Call Connect

' Start Tag Control Operation
Call controller.Execute("StartInventory")

' Stop Tag Control Operation
Call controller.Execute("Stop")
```

3.3.2.7. ClearBuffer

Clears the read prevention buffer twice.

Item	Type Description
Argument	Without
Returned value	Without

Examples of uses

```
' Connection
Call Connect
' Stop Tag Control Operation
Call controller.Execute("ClearBuffer")
```

3.4. Variable List

Defines a list of variables that are available for each class. Variables refer to objects of CaoVariable classes.

3.4.1. CaoController Classes Variables

Variable name	Description	Value		Reference
		Get	Put	
@MAKER_NAME	Get the manufacturer name.	✓	-	P.35
@VERSION	Get the provider version.	✓	-	P.35
@DEVICE_VERSION	Get the firmware version.	✓	-	P.35
@SERIAL	Get the serial number.	✓	-	P.36
@SERIAL_AND_PARTNO	Gets the part number and serial number.	✓	-	P.36
@BATTERYLEVEL	Acquires the remaining battery level.	✓	-	P.36
@TRIGGERMODE	Sets the trigger mode.	-	✓	P.37
@READTYPEOUTPUT	Sets/acquires Read outputs.	✓	✓	P.37
@WRITETYPEOUTPUT	Sets/acquires Write outputs.	✓	✓	P.38
@FREQUENCY	Sets/acquires frequency settings.	✓	✓	P.38
@QVALUE	Set/get Q value.	✓	✓	P.38
@SESSION	Set/Get Session.	✓	✓	P.39
@SESSIONINIT	Set/Get SessionInit.	✓	✓	P.39
@REREADPREVENTION	You can set/acquire the reading prevention setting twice.	✓	✓	P.40
@VERIFY	Set/Get Write verify.	✓	✓	P.40
@POWERSAVING	Sets/obtains the power saving setting.	✓	✓	P.40
@POLARIZATION	Sets/obtains the polarity setting.	✓	✓	P.41

@RESPONSE_FORMAT	Gets/sets the response format.	-	✓	P.41

3.4.1.1. @MAKER_NAME

Get the manufacturer name.

Data type

Type Description	
VT_BSTR	Get the manufacturer name.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@MAKER_NAME")
' Value acquisition
Dim strVal As String
StrVal = var.value
```

3.4.1.2. @VERSION

Get the provider version.

Data type

Type Description	
VT_BSTR	Get the provider version.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@VERSION")
' Value acquisition
Dim value As String
Value = var.value
```

3.4.1.3. @DEVICE_VERSION

Get the UR30 firmware version.

Data type

Type Description	
VT_BSTR	Get the firmware version of the UR30.

Examples of uses

```
' Connection
```

```
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@DEVICE_VERSION")
' Value acquisition
Dim value As String
Value = var.value
```

3.4.1.4. @SERIAL

Get the UR30 serial number.

Data type

Type	Description
VT_BSTR	Get the firmware version of the UR30.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@SERIAL")
' Value acquisition
Dim value As String
Value = var.value
```

3.4.1.5. @SERIAL_AND_PARTNO

Get the UR30 part number and serial number.

Data type

Type	Description
VT_BSTR	Get the firmware version of the UR30.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@SERIAL_AND_PARTNO")
' Value acquisition
Dim value As String
Value = var.value
```

3.4.1.6. @BATTERYLEVEL

Get the UR30 battery levels.

Data type

Type Description	
VT_I4	Get UR30 battery levels.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@BATTERYLEVEL")
' Value acquisition
Dim value As Long
Value = var.value
```

3.4.1.7. @TRIGGERMODE

Sets the trigger mode.

Data type

Type Description	
VT_I4	Sets the trigger mode.

Examples of uses

```
Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@TRIGGERMODE")
' Value acquisition
Dim value As Long
value = var.value
```

3.4.1.8. @READTYPEOUTPUT

Sets/acquires Read outputs.

Data type

Type Description	
VT_I4	Sets/acquires Read outputs.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@READTYPEOUTPUT")
' Value acquisition
Dim value As Long
```

```
value = var.value
' Value setting
var.value = 1
```

3.4.1.9. @WRITETYPEOUTPUT

Sets/acquires Write outputs.

Data type

Type	Description
VT_I4	Sets/acquires Write outputs.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@WRITETYPEOUTPUT")
' Value acquisition
Dim value As Long
value = var.value
' Value setting
var.value = 1
```

3.4.1.10. @FREQUENCY

Sets/acquires frequency settings.

Data type

Type	Description
VT_I4	Sets/acquires frequency settings.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@FREQUENCY")
' Value acquisition
Dim value As Long
value = var.value
' Value setting
var.value = 1
```

3.4.1.11. @QVALUE

Set/ acquires Q value.

Data type

Type	Description

VT_I4	Set/ acquires Q value.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@QVALUE")
' Value acquisition
Dim value As Long
value = var.value
' Value setting
var.value = 1
```

3.4.1.12. @SESSION

Sets/ acquires the session flag.

Data type

Type Description	
VT_I4	Sets/ acquires the session flag.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@SESSION")
' Value acquisition
Dim value As Long
value = var.value
' Value setting
var.value = 1
```

3.4.1.13. @SESSIONINIT

Sets/ acquires the session flag initialization setting.

Data type

Type Description	
VT_I4	Sets/ acquires the session flag initialization setting.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@SESSIONINIT")
' Value acquisition
Dim value As Long
value = var.value
' Value setting
```

```
    var.value = 1
```

3.4.1.14. @REREADPREVENTION

Sets/ acquires the reading prevention setting twice.

Data type

Type	Description
VT_I4	sets/ acquires the reading prevention setting twice.

Examples of uses

```
    ' Connection
    Call Connect
    ' Add Variable
    Dim var As CaoVariable
    Set var = controller.AddVariable("@REREADPREVENTION")
    ' Value acquisition
    Dim value As Long
    value = var.value
    ' Value setting
    var.value = 1
```

3.4.1.15. @VERIFY

Sets/ acquires the verify setting at the time of Write.

Data type

Type	Description
VT_I4	Sets/ acquires the verify setting at the time of Write.

Examples of uses

```
    ' Connection
    Call Connect
    ' Add Variable
    Dim var As CaoVariable
    Set var = controller.AddVariable("@VERIFY")
    ' Value acquisition
    Dim value As Long
    value = var.value
    ' Value setting
    var.value = 1
```

3.4.1.16. @POWERSAVING

Sets/ acquires the power saving setting.

Data type

Type	Description
VT_I4	Sets/ acquires the power saving setting.

Examples of uses

```
' Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@POWERSAVING")
' Value acquisition
Dim value As Long
value = var.value
' Value setting
var.value = 1
```

3.4.1.17. @POLARIZATION

Sets/ acquires the polarity setting.

Data type

Type	Description
VT_I4	Sets/ acquires the polarity setting.

Examples of uses

```
Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@POLARIZATION")
' Value acquisition
Dim value As Long
Value = var.value
' Value setting
Var.value = 1
```

3.4.1.18. @RESPONSE_FORMAT

Sets the response format.

Data type

Type	Description
VT_UI4|VT_ARRAY	Sets the response format.

Examples of uses

```
Connection
Call Connect
' Add Variable
Dim var As CaoVariable
Set var = controller.AddVariable("@RESPONSE_FORMAT")
' Value setting
Var.value = ARRAY(1, 1, 1, 1)
```

3.5. List of events

Notifies the information of the RF tag detected by the scanner. A message is issued once for a tag, and if multiple tags are read, the message is reported as many times as it has been read.

The content of the message varies depending on the RF tag control command. For details, see below.

Message number	Description	Reference
41	Notifies the information of the RF tag detected during the execution of the RF tag control by the StartInventory.	P.42
42	Notifies the information of the RF tag detected during the RF tag control by the StartRead and the data read from the specified memory area.	P.43
43	Notifies the information of the RF tag detected during the execution of the RF tag control by the StartWrite.	P.44
44	Notifies the information of the RF tag detected during the execution of the RF tag control by the StartLock.	P.45
45	Notifies the information of the RF tag detected during the execution of the RF tag control by the StartKill.	P.46

3.5.1. Inventory messages

The scanners notify the information of the detected RF tag during the RF tag control by the StartInventory (see 3.3.2.1).

Item	Description		
Message number	41		
Message content	VT_ARRAY | VT_VARIANT		
	1	VT_BSTR	UII data of the detected RF tag
	2	VT_BSTR Or VT_EMPTY	PC data of the detected RF tag[4]
	3	VT_BSTR Or VT_EMPTY	RSSI4 for detecting RF-tags[4]
	4	VT_BSTR Or VT_EMPTY	Antenna No. 4 used for RF tag detection[4]
	5	VT_BSTR Or VT_EMPTY	Polarisation State 4 during RF Tag Detection[4]

[4] Returns data only when permitted in response format setting, or VT_EMPTY if prohibited

3.5.2. Read messages

The scanners notify the information of the RF tag detected during the RF tag control by the StartRead (see 3.3.2.2).When the RF tag communication result is normal, the specified memory area is also notified.

Item	Description		
Message number	42		
Message content	VT_ARRAY | VT_VARIANT		
	1	VT_BSTR	UII data of the detected RF tag
	2	VT_BSTR Or VT_EMPTY	PC data 4 of the detected RF tag[4]
	3	VT_BSTR Or VT_EMPTY	RSSI4 for detecting RF-tags[4]
	4	VT_BSTR Or VT_EMPTY	Antenna No. 4 used for RF tag detection[4]
	5	VT_BSTR Or VT_EMPTY	Polarisation State 4 during RF Tag Detection[4]
	6	VT_BSTR	RF Tag Communication Result "0000": Normal termination "0203": Access outside the memory range "0204": Access to lock memory "020F": Other error
	7	VT_BSTR or VT_UI1 | VT_ARRAY	Read data Byte-ordering or character-stringing depends on the values specified by the parameters at StartRead time.

3.5.3. Write messages

The scanner notifies the scanner of the RF tag information detected during the RF tag control by the StartWrite (see 3.3.2.3).If the RF tag communication result is normal, the scanner writes the information to the specified memory area.

Item	Description		
Message number	43		
Message content	VT_ARRAY | VT_VARIANT		
	1	VT_BSTR	UII data of the detected RF tag
	2	VT_BSTR Or VT_EMPTY	PC data 4 of the detected RF tag[4]
	3	VT_BSTR Or VT_EMPTY	RSSI4 for detecting RF-tags[4]
	4	VT_BSTR Or VT_EMPTY	Antenna No. 4 used for RF tag detection[4]
	5	VT_BSTR Or VT_EMPTY	Polarisation State 4 during RF Tag Detection[4]
	6	VT_BSTR	RF Tag Communication Result "0000": Normal termination "0203": Access outside the memory range "0204": Access to lock memory "020F": Other error

3.5.4. Lock messages

The scanner notifies the scanner of the information of the RF tag detected during the RF tag control by the StartLock (see 3.3.2.4). When the RF tag communication result is normal, the lock status of the specified memory area is changed to the detected tag.

Item	Description		
Message number	44		
Message content	VT_ARRAY | VT_VARIANT		
	1	VT_BSTR	UII data of the detected RF tag
	2	VT_BSTR Or VT_EMPTY	PC data 4 of the detected RF tag[4]
	3	VT_BSTR Or VT_EMPTY	RSSI4 for detecting RF-tags[4]
	4	VT_BSTR Or VT_EMPTY	Antenna No. 4 used for RF tag detection[4]
	5	VT_BSTR Or VT_EMPTY	Polarisation State 4 during RF Tag Detection[4]
	6	VT_BSTR	RF Tag Communication Result "0000": Normal termination "0203": Access outside the memory range "0204": Access to lock memory "020F": Other error

3.5.5. Kill messages

The scanners notify the detected RF tags during StartKill RF tag control (3.3.2.5).If the RF tag communication result is normal, the detected tags are turned Kill and cannot be used.

Item	Description		
Message number	45		
Message content	VT_ARRAY | VT_VARIANT		
	1	VT_BSTR	UII data of the detected RF tag
	2	VT_BSTR Or VT_EMPTY	PC data 4 of the detected RF tag[4]
	3	VT_BSTR Or VT_EMPTY	RSSI4 for detecting RF-tags[4]
	4	VT_BSTR Or VT_EMPTY	Antenna No. 4 used for RF tag detection[4]
	5	VT_BSTR Or VT_EMPTY	Polarisation State 4 during RF Tag Detection[4]
	6	VT_BSTR	RF Tag Communication Result "0000": Normal termination "0203": Access outside the memory range "0204": Access to lock memory "020F": Other error

4. Programming by UR30 providers

For UR30 providers, you can connect your clients to scanners as follows:

- Creating a CaoEngine
- Creating a CaoWorkspace
- Creating a CaoController

After you connect to the scanner, you can access the scanner's info by using the CaoController's Execute method or by generating CaoVariabe objects.

4.1. Sample Programming Connecting to a Scanner and Reading UII Data on RF Tags

The following example shows a sample program that sets UR30 parameters and saves them to the flash memory. Table 4-1 describes the requirements of the sample program and 4.1.1 describes the flow of the sample program.

Table 4-1 Sample Program Requirements

Requirement	Description
Destination	COM
	The host COM number is 3, and other items are omitted.
Processing content	Connecting to Scanners with Continuous Mode 2, PC Data Transfer Enable, and RSSI Data Transfer Enable
	Starting Communication Control with Scanners in Inventory Modes
	Displays a message when UII data is read from the scanner.
	Disconnect after displaying the message

4.1.1. Sample program

The following is an overview of the sample program.

Samples are described in VB.Net.

Sample	SampleInventory.vb

```
' Object
Dim engine As CaoEngine
Dim workspace As CaoWorkspace
Dim WithEvents controller As CaoController
Dim MessageBuf As Object

' Connection method
Private Sub Connect()
    ' Creating CaoEngine Objects
    Engine = New CaoEngine
    ' Creating CaoWorkspace Objects
    Workspace = engine.AddWorkspace("Newwrks", "")
    ' Creating CaoController Objects
    Controller = workspace.AddController("SampleController", _
                                                    "CaoProv.DENSO.UR30",_
                                                    "",_
                                                    "conn=COM:3, ResponseFormat=(1, 1, 0, 0)")
    ' Create a OnMessage Event Handler
    AddHandler controller.OnMessage, AddressOf controller_OnMessage
End Sub

' Disconnect method
Private Sub Disconnect()
      ' Send stop commands
      If controller IsNot Nothing Then
                Call controller.Execute("stop")
       End If

      ' Deletes OnMessage handlers
      RemoveHandler controller.OnMessage, AddressOf controller_OnMessage
      ' Remove CaoController from CaoWorkspace
      Call workspace.Controllers.Remove(controller.Index)
      ' Clear CaoController
      System.Runtime.InteropServices.Marshal.ReleaseComObject(controller)
      Controller = Nothing
      ' Remove CaoWorkspace from CaoEngine
      Call engine.Workspaces.Remove(workspace.Index)
      ' Clear CaoWorkspace
      System.Runtime.InteropServices.Marshal.ReleaseComObject(workspace)
      Workspace = Nothing
      ' Clear CaoEngine
      System.Runtime.InteropServices.Marshal.ReleaseComObject(engine)
      Engine = Nothing
End Sub

' Starting Inventory Modes
Private Sub StartInventory()
    ' Connection
    Call Connect()
    ' Starting Inventory Modes
    Controller.Execute("StartInventory")
End Sub

' Notify Message Events from Providers
Private Sub controller_OnMessage(ByVal pICaoMess As CAOLib.CaoMessage)
    MessageBuf = pICaoMess.Value
    ' UII data
    Dim uiiData As String
    UiiData = MessageBuf(0)
    ' PC data
    Dim pc As String
    Pc = MessageBuf(1)
    ' Data RSSI
    Dim rssi As String
    Rssi = MessageBuf(2)
    ' Used antenna
    Dim antenna As String
    Antenna = MessageBuf(3)
    ' Polarisation status
```

```
        Dim polarization As String
        Polarization = MessageBuf(4)

        ' Show 'UII Data in Message Box
        MsgBox(uiiData)
    End Sub
```

4.1.1.1. Connection

To connect to the scanner, perform the following steps:

(1) Provides variables for holding objects. The objects required for connecting controllers are the CaoEngine object, the CaoWorkspace object, and the CaoController object. The CaoWorkpace object does not need to have variables when retrieving the CaoController object from the CaoWorkspaces. You can also name variables with a WithEvents to receive OnMessage events from the CaoController.

```
Dim engine As CaoEngine                      ' Variables for CaoEngine Objects
Dim workspace As CaoWorkspace                ' Variables for CaoWorkspace Objects
Dim WithEvents controller As CaoController   ' Variables for CaoController Objects
```

(2) Generates CaoEngine objects. CaoEngine objects are generated using the new keywords.

```
' Creating CaoEngine Objects
Set engine = New CaoEngine
```

(3) Gets or creates a CaoWorkspace object. When you create a CaoEngine object, it defaults to creating one CaoWorkspaces object and one CaoWorkspace object. Here are some examples of newly generated CaoWorkspace objects and a CaoWorkspace of defaults:

```
' Creating CaoWorkspace Objects
Set workspace = engine.AddWorkspace("NewWrks", "")
```

(4) Generate CaoController objects. To generate CaoController objects, set the provider name to be used and the parameters to be used. For UR30 provider, specify optional response formats in advance when connecting to scanners. Here is an example of the code:

```
' Creating CaoController Objects
Set controller = workspace.AddController("SampleController", _
                                         "CaoProv.DENSO.UR30", _
                                         "", _
                                         "conn=COM:3 ,ResponseFormat=(1, 1, 0, 0))
```

4.1.1.2. UR30 is executed in the UII-data retrieval mode and the UII-data retrieval mode is set to the UII-data retrieval mode.

Starts the UR30 in UII-acquisition mode (Inventory).

```
' Starting Inventory Modes
Controller.Execute("StartInventory")
```

4.1.1.3. Receive a message when reading a tag

When Tags information is received from the UR30, a OnMessage event is issued. When a OnMessage event is received, UII data is displayed in MsgBox from the content of the message.

```
Private Sub controller_OnMessage(ByVal pICaoMess As CAOLib.CaoMessage)
    MessageBuf = pICaoMess.Value

     ' The UII data stored in the received data is displayed in the message box.
     MsgBox MessageBuf(0)

End Sub
```

4.1.1.4. Disconnection

When disconnecting from the controller, delete the created object and delete the object to be deleted from the collection class that manages the object. Here is an example of the code:

```
          ' Deletes OnMessage handlers
          RemoveHandler controller.OnMessage, AddressOf controller_OnMessage
          ' Remove CaoController from CaoWorkspace
          Call workspace.Controllers.Remove(controller.Index)
          ' Clear CaoController
          System.Runtime.InteropServices.Marshal.ReleaseComObject(controller)
          Controller = Nothing
          ' Remove CaoWorkspace from CaoEngine
          Call engine.Workspaces.Remove(workspace.Index)
          ' Clear CaoWorkspace
          System.Runtime.InteropServices.Marshal.ReleaseComObject(workspace)
          Workspace = Nothing
          ' Clear CaoEngine
          System.Runtime.InteropServices.Marshal.ReleaseComObject(engine)
          Engine = Nothing
```

5. UR30 provider error code

The following unique error codes masked with the 0x8011 **** are available for this service provider:. (Refer to Table 5-1 Unique Error Code Table)

Refer to the Error Codes section of the ORiN2 Programming Guide (Link) for information about ORiN2 common errors.C:¥ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_en.pdf

Table 5-1 Unique Error Code Table

Error Number	Description
0x80110002	Receive unexpected data
0x80110003	Trigger mode specification error Incorrect TriggerMode specification.
0x80110004	Response format error The setting of the response format of the UR30 is different from the setting of the response format held by the providers. Run the SetResponseFormat again.
0x8010A300	Operation mode error
0x8010A301	Execution condition error
0x8010A303	Format error
0x8010A304	Out of Argument Range Error
0x8010AA00	Charging error
0x8010AA01	Trigger mode error
0x8010****	Unexpected error

It also returns the UR30 error code masked with "0x8010 ****".

Appendix A. Communication protocol command correspondence table

CaoWorkspace

Method	Communications command
AddController	RFUS
	RFUU
	RFURF2
Controllers::Remove	RFUS

CaoController::Execute

Command	Communications command
GetVersion	VER
GetSirial	ID
GetSirialAndPartno	IDF
GetBatteryLevel	VBAT
SetTriggerMode	RFUU
SetReadTypeOutput	RFUPS
GetReadTypeOutput	RFUPG
SetWriteTypeOutput	RFUPS
GetWriteTypeOutput	RFUPG
SetFrequency	RFUPS
GetFrequency	RFUPG
SetQValue	RFUPS
GetQValue	RFUPG
SetSession	RFUPS
GetSession	RFUPG
SetSessionInit	RFUPS
GetSessionInit	RFUPG
SetRereadPrevention	RFUPS
GetRereadPrevention	RFUPG
SetVerify	RFUPS
GetVerify	RFUPG

SetPoewerSaving	RFUPS
GetPoewerSaving	RFUPG
SetPolarization	RFUPS
GetPolarization	RFUPG
SaveParameter	RFUPW
StartInventory	RFUI
	RFUIU
StartRead	RFUR
	RFURU
StartWrite	RFUW
	RFUWU
StartLock	RFUL
	RFULU
StartKill	RFUK
	RFUKU
Stop	RFUS
ClearBuffer	RFUC
SetResponseFormat	RFURF2