

# PATLITE WDR Provider

Version 1.0.0

User's Guide

August 3, 2022

Remark

\* The images of WDR and WDT in this manual were provided by PATLITE CORPORATION.

## Revision history

Version	Dating	Content
1.0.0	2019-02-01	First edition
	2020-07-22	Added operation-checked device and a combination of compatible models.
	2022-08-03	Provider name error corrected.

## Compatible device

Model	Version	Note
WDR-L-Z2	1.00 ~	AC adapter attached
WDR-LE-Z2	1.00 ~	No AC adapter (for overseas use)
WDR-L-Z2-PRO	1.00 ~	Included with domestic AC adapter
WDR-LE-Z2-PRO	1.00 ~	No AC adapter for overseas use
WDR-L-Z2-PRO-L	1.00 ~	(Lite version) Included with domestic AC-Adapter
WDR-LE-Z2-PRO-L	1.00 ~	(Lite version) No AC-adapter for foreign use

## Operation Check Device

Model	Version	Notes
WDR-L-Z2	2.04	

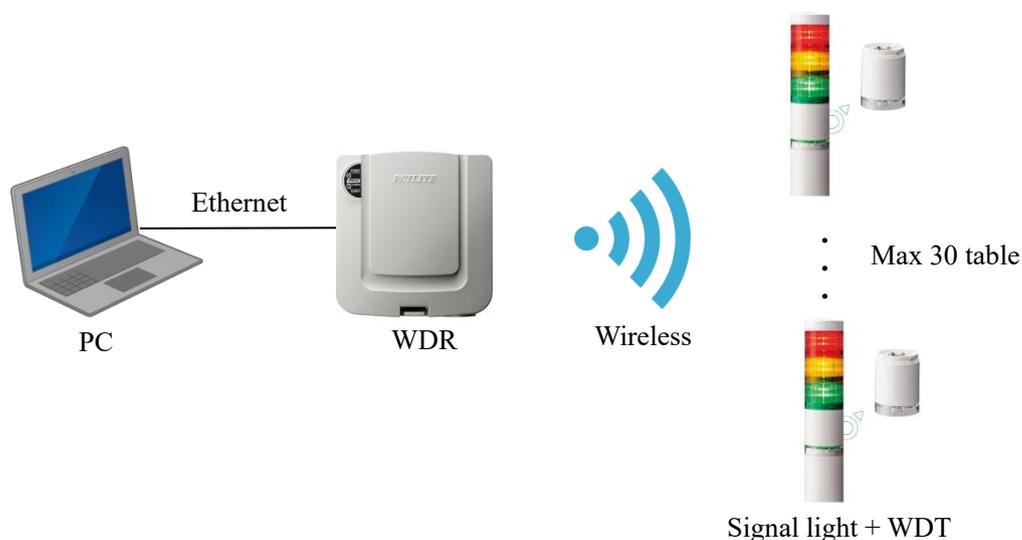
---

**Table of contents**

1. Introduction .....	4
2. Combination of Compatible Models .....	4
3. Environment setup.....	5
4. Provider overview .....	8
4.1. Overview .....	8
4.1.1. WDT state transition.....	9
4.1.2. How to use the transmission mode.....	10
4.2. Method Properties .....	14
4.2.1. CaoWorkspace::AddController method .....	14
4.2.2. CaoController::get_VariableNames properties.....	16
4.2.3. CaoController::AddVariable method.....	17
4.2.4. CaoVariable::get_Value properties .....	18
4.3. Error-code.....	20
5. Sample code(CaoScript).....	21

# 1. Introduction

This document is a user's guide for WDR providers who obtain signaling light information from PATLITE's WDR-L-Z2, WDR-LE-Z2 (hereinafter referred to as "WDR"). WDR is a receiver that can wirelessly receive multiple signaling light information. Fig. 1-1 shows the environmental configuration of WDR providers. The PC-WDR is connected by Ethernet cables, and the WDR and signaling lights are wirelessly communicated. One WDR can obtain up to 30 signal light information. The signaling light must be equipped with a WDT-6LR-Z2, WDT-5LR-Z2, WDT-6M-Z2 or WDT-5M-Z2 (hereinafter referred to as "WDT"), which is a data transmitter.



**Fig. 1-1 Environment Configuration of WDR Providers**

# 2. Combination of Compatible Models

Table 2-1 shows the combinations of receivers and transmitters supported by WDR provider. WDR provider cannot communicate with transmitter WDT-PRO series. If you want to use WDT-PRO series, use WDR-PRO provider which is the successor provider.

**Table 2-1 WDR Providers Correspondence Table**

Receiver \ Transmitter	WDT Series	WDT-PRO Series
WDR Series	○	×
WDR-PRO Series	○	×

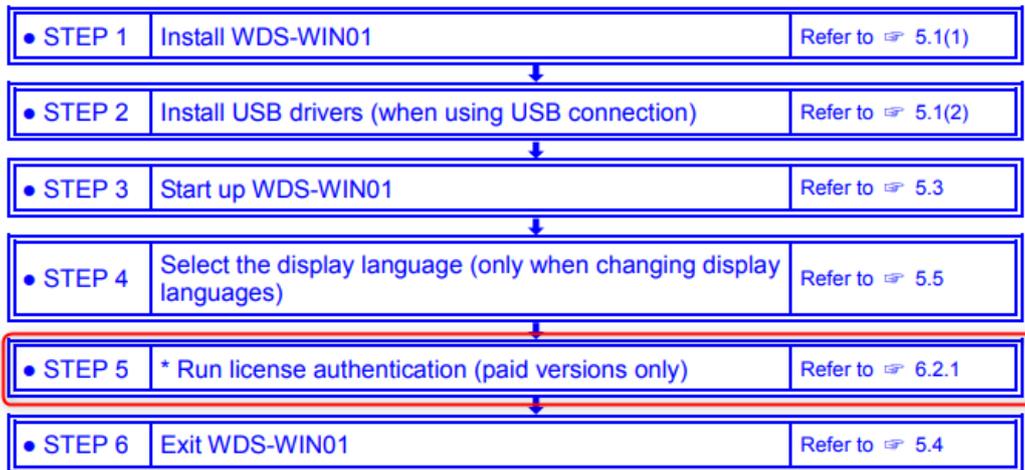
### 3. Environment setup

To use a WDR provider, you must set up WDR and WDT. For setup, please use WDS-WIN01, a software for operating the Patrite system. WDS-WIN01 can be DLed from Mipage on the Patrite HP.

See "Comprehensive Instruction Manual for MODEL:WDS-WIN01 of Operating Software for Wireless Data Communication System" for instructions on how to set up the software. Complete the setup by following the steps of "4. Using the 1 WDS-WIN01 for the first time" and "4. Configuring 2 WDT and WDRs".

If you are using only WDR providers, you do not need the STEP5 licensing procedures in Section 4.1.

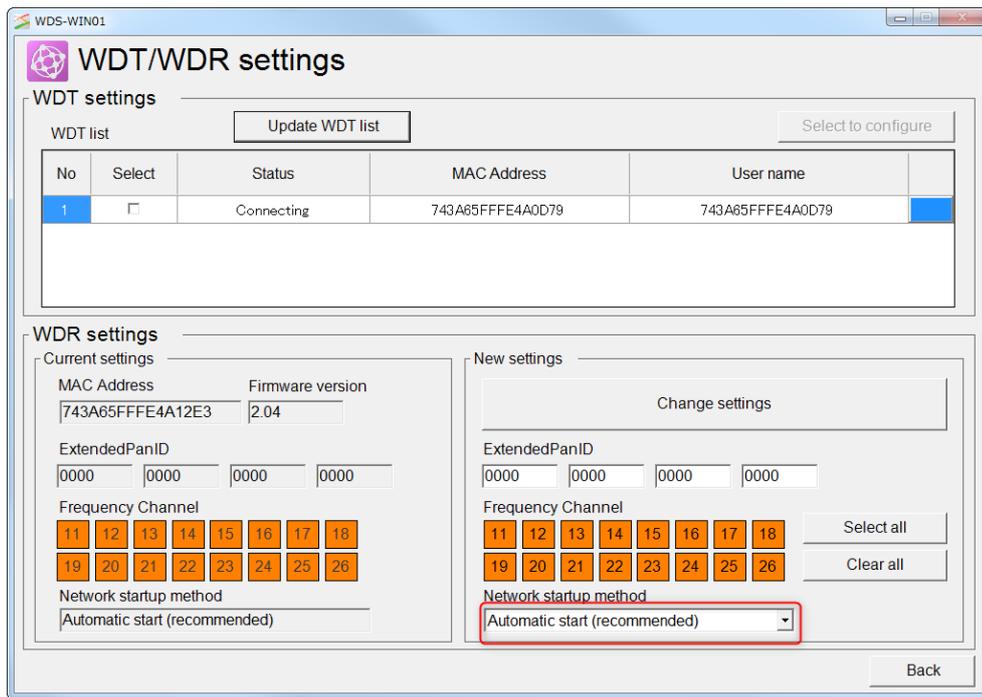
#### 4.1 Using WDS-WIN01 for the first time



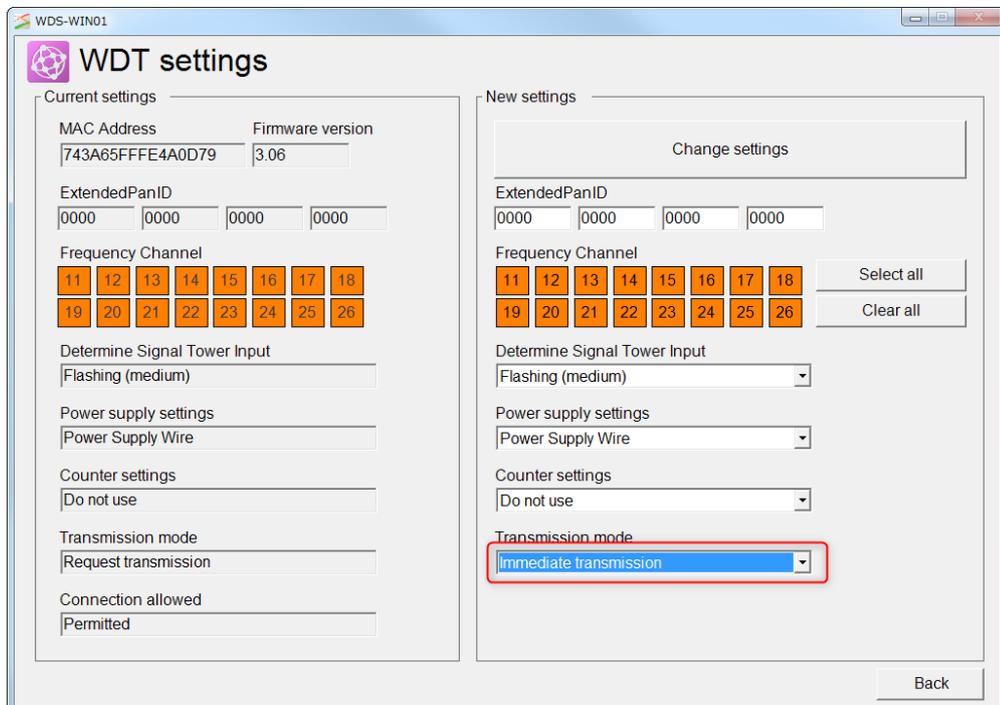
1

In Chapter 4.2, "Configuring the STEP5 WDT," be sure to select "Automatic start (recommended)" to start the WDR network. If "Manual Start" is selected, the WDR provider and WDR cannot be connected.

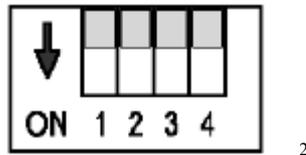
<sup>1</sup> PATLITE Corporation WDS-WIN01 Instruction Manual



The WDT transmission mode can be either "Immediat transmission" or "Request transmission ". See Section 3.1.2 of this Provider Manual for more information.



The WDT-6LR-Z2, WDT-5LR-Z2 can add buzzer information in addition to signal light information. To add buzzer information, turn on the setting switch 3 located on the bottom of the WDT-6LR-Z2, WDT-5LR-Z2 shown in Fig. 3-1.



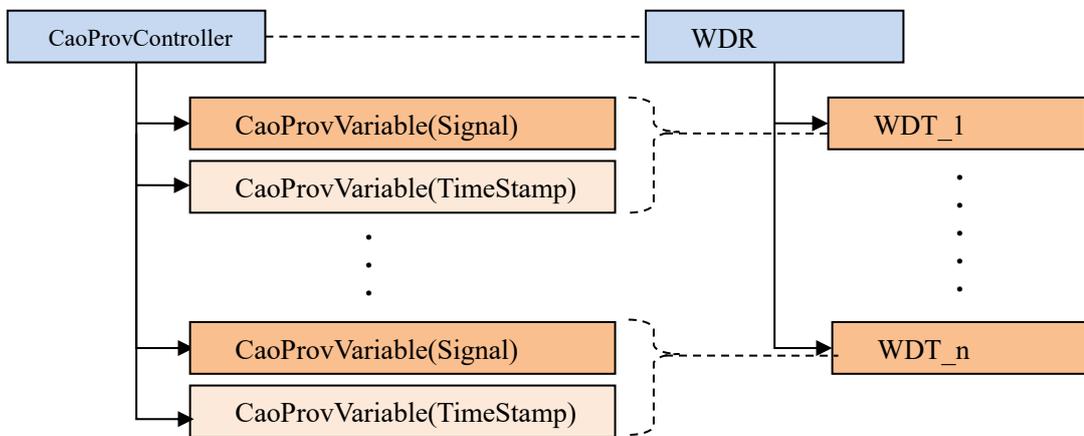
**Fig. 3-1 Setup switch**

<sup>2</sup> PATLITE Corporation WDT-5LR-Z2 , WDT-6LR-Z2, WDR-L(E)-Z2 Instruction Manual

## 4. Provider overview

### 4.1. Overview

Fig. 4-1 shows the correspondence between classes of providers and WDR and WDT. CaoProvController is a WDR, and CaoProvVariable is WDT. Only one CaoProvController can be connected to one WDR. CaoProvVariable has a Signal variable that stores the status of the signal light and a TimeStamp variable that stores the time of the signal light. Create a TimeStamp variable is optional. Be sure to create a Signal variable. If only the TimeStamp variable is created, the time of the TimeStamp variable is not updated because the WDR provider does not require signal light information at regular intervals, as described in chapter 3.1.2.



**Fig. 4-1 shows the correspondence between each class of provider and WDR**

To create the Signal and TimeStamp variables, use the WDT Mac address, which is listed in the device labels (Fig. 4-2).



**Fig. 4-2 WDT Mac Address**

<sup>3</sup> PATLITE Corporation WDT-5LR-Z2 , WDT-6LR-Z2, WDR-L(E)-Z2 Instruction Manual

The file format of WDR providers is DLL (Dynamic Link Library). Details are shown in Table 4-1.

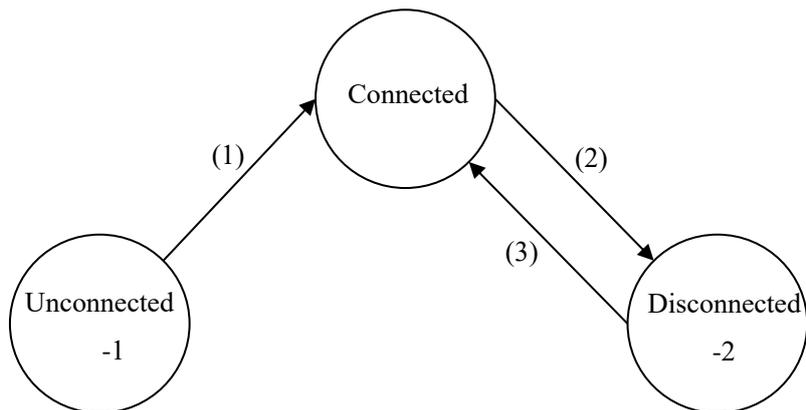
**Table 4-1 WDR Provider**

File name	CaoProvPatliteWDR.dll
ProgID	CaoProv.PATLITE.WDR
Registry registration <sup>4</sup>	Regsvr32 CaoProvPatliteWDR.dll
Deletion of Registry Registration	Regsvr32 /u CaoProvPatliteWDR.dll

**4.1.1. WDT state transition**

WDT transitions to three states: unconnected, connected, and disconnected. Before WDT joins the WDR network, WDT is in a unconnected state. In this state, signal light information cannot be obtained from WDT. When WDT joins the network, WDT enters a connected state and WDT signal light information can be obtained (1). When the WDT and WDR are unable to communicate with each other, the provider considers that an error has occurred in the WDT. The provider transitions from the connected state to the disconnected state. (2) The WDT recovers from the abnormal state. When the provider joins the WDR network again, the provider returns to the connected state. (3) The state of the WDT is determined by the values of the Signal variables. Returns "-1" when unconnected or "-2" when disconnected. Refer to 3.2.4 for the connection status.

Executing AddController in chapter 3.2.1 will reset the WDR's network. As a result, immediately after executing AddController, all WDTs are unconnected. It may take up to 100 seconds for the WDT to become connected from the unconnected state.



**Fig. 4-3 State Transition Diagram of WDT**

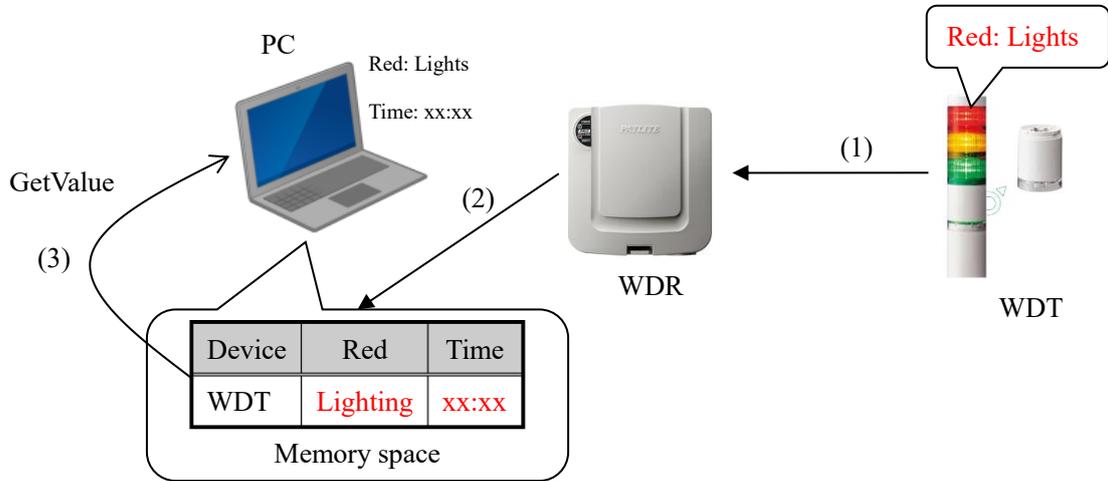
<sup>4</sup> You do not need to manually register/delete ORiN SDK installations.

**4.1.2. How to use the transmission mode**

The WDR provider has two ways to use depending on the transmission mode of the WDT. Select "Immediate transmission mode" to know the latest state of the signal light, or "Request transmission mode" to monitor the signal information only in the specified cycle.

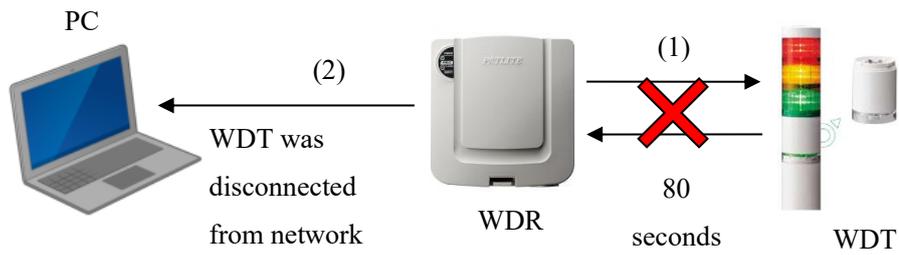
**Immediate transmission mode**

Fig. 4-4 shows how to use the WDR provider when the WDT is in immediate transmission mode. In immediate transmission mode, the WDT transmits signal data to WDR (1). When reception is completed, the WDR transmits the received data to the PC, and the signal light information and the time the PC received the data in the memory space of the WDR provider are recorded (2). When GetValue is executed, the WDR provider acquires and displays the signal lamp information recorded in the memory space and the reception time (3).



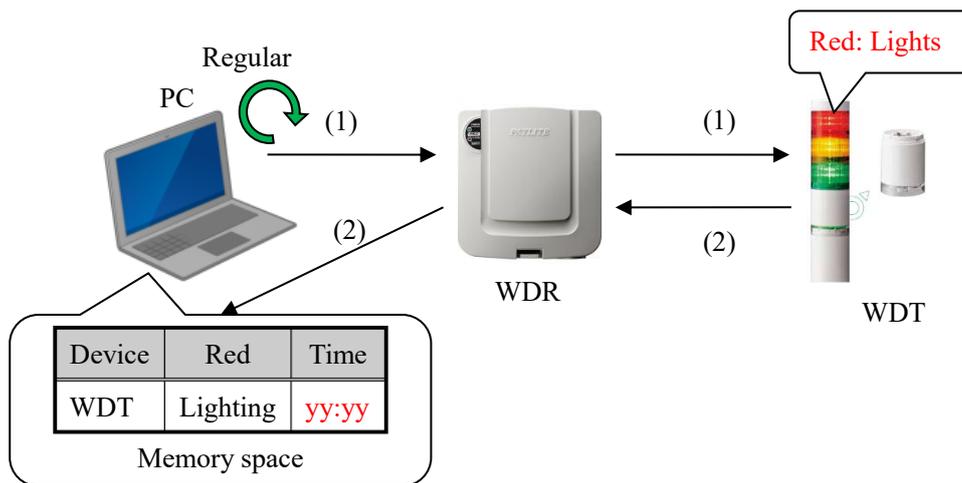
**Fig. 4-4 Using WDR Provider in Immediate transmission mode**

In the immediate transmission mode, the signal information is transmitted at the timing when the signal light changes, so that it cannot be recognized even if the WDT is disconnected. Therefore, the WDR periodically communicates with the WDT to monitor whether the WDT is in a normal state. Fig. 4-5 shows the survival checking function between the WDR-WDT. When the communication between WDR and WDT is interrupted for 80 seconds (1), the WDT determines that the WDT is disconnected from the network and notifies the WDR provider (2).



**Fig. 4-5 Survival-checking function between WDR-WDT**

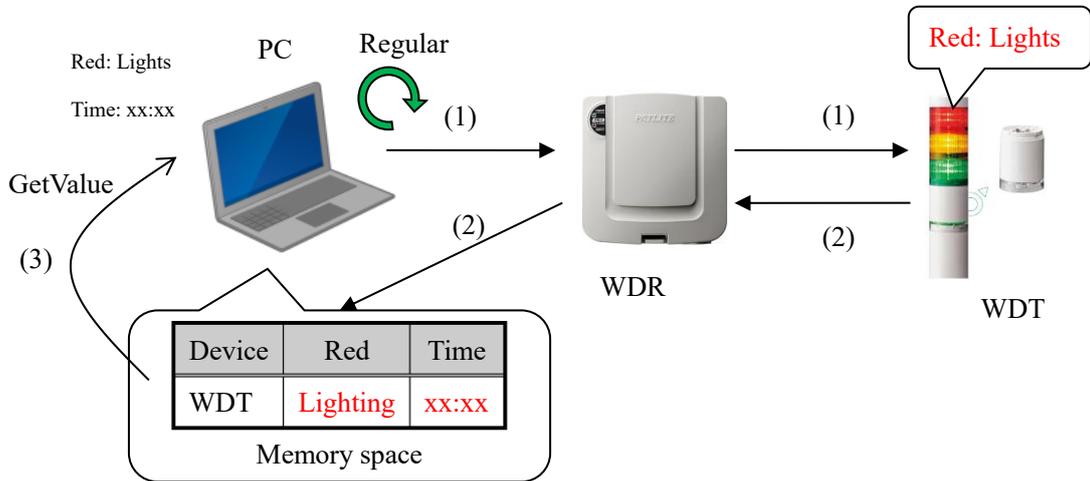
You can monitor whether WDT is normal by using the WDR-WDT survival check feature, but if you want to know the WDT's abnormal condition earlier, use the WDR-provider survival check feature. Fig. 4-6 shows the survival check function between WDR providers and WDT. WDR providers periodically make WDT and survival check requests via WDR to monitor whether WDT is in a normal state (1) and returns a response when WDT receives the request normally to update the reception time of the memory space (2). If there is no response for 17 seconds after a survival confirmation request, the request is considered to have failed. When the number of failures reaches a specified number, the WDT is considered to be in an abnormal state and it is determined that the WDT has left the network. The specified number of failures can be specified in the FailureCount options at AddController. Also, you can specify the period at which you issue a confirm-alive request by using the Interval option at the time of AddController. This function is not executed by specifying "-1" in the Interval option.



**Fig. 4-6 Survival Verification Function between WDR Provider and WDT**

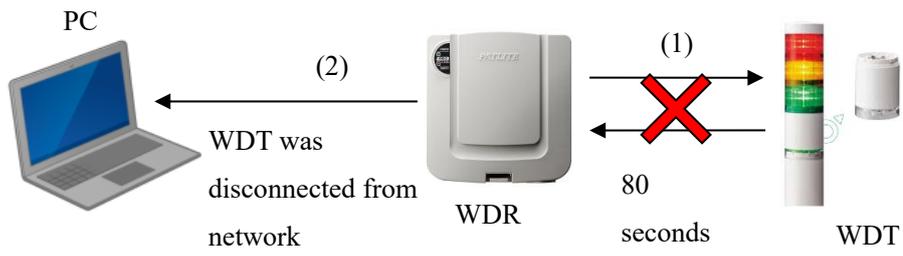
**Request transmission mode**

Fig. 4-7 shows how to use the WDR provider when the WDT is in request transmission mode. In request transmission mode, signal light information is not transmitted when changing. Therefore, the WDR provider periodically requests WDT signal information via WDR (1). When the WDT receives the request, it returns the signal information at that time, and the signal light information and the time the PC received in the PC memory space are recorded (2). When GetValue is executed, the WDR provider acquires and displays the signal lamp information recorded in the memory space and the reception time (3). If there is no response for 17 seconds after requesting the signal information, the request is considered to have failed. When the failure reaches the prescribed number, the WDT considers it to be in an abnormal state and judges that it has left the network. The specified number of failures can be specified with the FailureCount option at AddController. The interval for requesting information can be specified with the Interval option at AddController. When using in request transmission mode, specify Interval option within the range of 500 to 2147483647 milliseconds.



**Fig. 4-7 How to Use WDR Provider in Request transmission mode**

In request transmission mode, the WDR periodically communicates with the WDT to monitor whether the WDT is in the normal state. Fig. 4-8 shows the survival checking function between the WDR-WDT. When the communication between WDR and WDT is interrupted for 80 seconds (1), the WDT will notify the WDR provider of the decision that the WDT has been disconnected from the network (2) If the WDR provider judges that the WDT has already disconnected from the network, this notification will not be made.



**Fig. 4-8 Survival-checking function between WDR-WDT**

## 4.2. Method Properties

### 4.2.1. CaoWorkspace::AddController method

In this provider, the CaoWorkspace::AddController sets the connection parameters to connect the provider. The AddController specifications are shown below. All WDTs are unconnected state immediately after the AddController is executed. Information on the IP address of the WDR set in chapter 2 is required

**Form**

```

AddController(
    BSTR bstrCtrlName,           //[in] controller name
    BSTR bstrProvName,          //[in] provider name
    BSTR bstrPcName,            //[in] provider's execution machine name
    BSTR bstrOption              //[in] option string
);
    
```

Argument	Description			
BstrCtrlName	Enter any controller name. e.g., "Ctrl1"			
BstrProvName	Enter "CaoProv.PATLITE.WDR".			
BstrPcName	Enter the name of the provider's running machine. If the provider is local, specify a blank.			
BstrOption	You can specify the connection destination and WDT survival confirmation signal transmission interval.			
	Setting items	Option String	Required	Description
	Destination	Conn	○	Specify the IP address and port number of the connected device. Enter in the following format. The parameter in brackets ("[]") is optional. If you do not specify an option, the destination port defaults to 10001.  Eth:<destination IP>[:<destination port>[:<local IP>[:<local port>]]]"
	WDT signal information request interval	Interval	-	When the WDT mode is the immediate transmission mode, the interval is the survival confirmation request interval, and when the WDT mode is the request transmission mode,

(WDT survival check request interval)			<p>the interval is the signal information request interval. Specify the transmission interval in milliseconds between 500 and 2147483647 milliseconds. When -1 is specified, this function is not executed. If you do not specify a value, 20000 milliseconds is the default value.</p> <p><b>NOTE: In request transmission mode, -1 must not be specified, but must be specified in the range of 500 to 2147483647 milliseconds.</b></p>
Number of transmission request failures	FailureCount	-	<p>Specify the number of transmission request failures to determine that an abnormal WDT is leaving. Specify in the range of 1 to 2147483647 times. Otherwise, the default value is three times.</p>
<p>Examples: "Conn = Eth: 192.168.0.1:10001"  "Conn = Eth:192.168.0.1:10001, Interval = 1000"  "Conn = Eth:192.168.0.1:10001, Interval = 1000, FailureCount = 1"</p>			

**Examples of Use (CaoScript)**

Dim ctrl

Set ctrl = cao.AddController("ctrl1", "CaoProv.PATLITE.WDR", "",

"Conn = Eth:192.168.0.1:10001, Interval = 1000, FailureCount = 1")

### 4.2.2. CaoController::get\_VariableNames properties

Gets the MAC address of the WDT that has been connected to the WDR. After executing the AddController method, it takes up to 100 seconds for the WDT to connect to the WDR. Returns the Empty type if there are no connected WDTs.

#### Form

```
Get_VariableNames(
    BSTR bstrOption           //[in] option string
);
```

Argument	Description
BstrOption	Not used. Specify a blank.

Type of return value	Description
ARRAY   BSTR	Returns the MAC address of a WDT connected to a WDR, "TimeStamp_" + MAC address" in the BSTR array. Returns the Empty type if there are no connected WDTs.

#### Examples of Use (CaoScript)

```
Dim varNamesList
VarNamesList = dat.ToVar(ctrl.VariableNames)
Dbg.output dat.ToVar(ctrl.VariableNames) (0) '⇒Display result example: 743a65ffe4a0d79
Dbg.output dat.ToVar(ctrl.VariableNames)(1) '⇒Display result example: TimeStamp_743a65ffe4a0d79
```

### 4.2.3. CaoController::AddVariable method

Creates CaoVariable objects that retrieve WDT signals.

\* Notes on Using TimeStamp Variables

Be sure to create a Signal variable with a TimeStamp variable. If you create only a TimeStamp variable, the date and time will not be updated because you do not require signaling lights at regular intervals.

#### Form

```
AddVariable (
    BSTR bstrVarName,           //[in] variable name
    BSTR bstrOption            //[in] option string
);
```

Argument	Description
BstrVarName	Specify the WDT MAC address or "TimeStamp_" + MAC address. If you specify the MAC address, the WDT generates a variable (Signal variable) that gets the WDT signal. If you specify the "TimeStamp_" + MAC address, then you can use the values that you obtained in.get_VariableNames to generate a variable (TimeStamp variable) that gets the date and time that the WDT signal was updated. MAC address is not case sensitive  Example 1) 743a65ffe4a0d79 MAC address Example 2) TimeStamp_743a65ffe4a0d79 "TimeStamp_" + MAC Address
BstrOption	Not used. Specify a blank.

#### Examples of Use (CaoScript)

```
Dim varSignal
Set varSignal = cao.Controllers("ctrl1").AddVariable(varNamesList (0), "")
Dim varTimeStamp
Set varTimeStamp = cao.Controllers("ctrl1").AddVariable(varNamesList (1), "")
```

#### 4.2.4. CaoVariable::get\_Value properties

Gets the states of the variables that are created by the AddVariable method. The return values depend on the Signal and TimeStamp variables. Refer to the table of returned values for details.

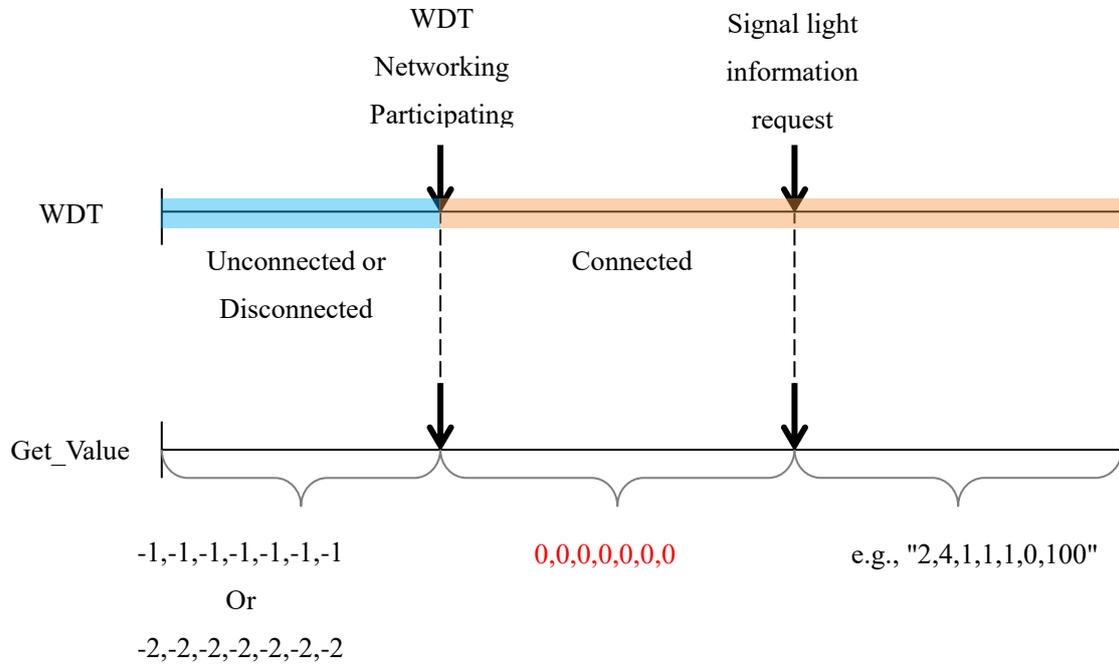
**Form**

Get\_Value ();

Argument	Description
Without	-

Return value	Type	Description																
Signal variables	ARRAY   I4	<p>Array elements return seven types of I4. Elements indicate "Red, Yellow, Green, Blue, White, Buzzer, LQI" from the left. Depending on the state, each value is shown in the following table:</p> <p>* LQI is a value that represents the quality of wireless communications. This is expressed as a number between 0 and 255.</p> <table border="1"> <thead> <tr> <th>State</th> <th>Color information</th> <th>Buzzer information</th> <th>LQI</th> </tr> </thead> <tbody> <tr> <td>Unconnected</td> <td>-1</td> <td>-1</td> <td>-1</td> </tr> <tr> <td>Connected</td> <td>0: Unused 1: Turning OFF 2: Lighting 4: Blinking</td> <td>0: Mute 1: Sound</td> <td>0-50: bad 51-100: Slight bad 101~150: Good 151~255: Best</td> </tr> <tr> <td>Disconnected</td> <td>-2</td> <td>-2</td> <td>-2</td> </tr> </tbody> </table>	State	Color information	Buzzer information	LQI	Unconnected	-1	-1	-1	Connected	0: Unused 1: Turning OFF 2: Lighting 4: Blinking	0: Mute 1: Sound	0-50: bad 51-100: Slight bad 101~150: Good 151~255: Best	Disconnected	-2	-2	-2
State	Color information	Buzzer information	LQI															
Unconnected	-1	-1	-1															
Connected	0: Unused 1: Turning OFF 2: Lighting 4: Blinking	0: Mute 1: Sound	0-50: bad 51-100: Slight bad 101~150: Good 151~255: Best															
Disconnected	-2	-2	-2															
TimeStamp variables	DATE	<p>Returns the DATE type. The TimeStamp variables show the values in the following tables, depending on the conditions.</p> <table border="1"> <thead> <tr> <th>State</th> <th>TimeStamp variables</th> </tr> </thead> <tbody> <tr> <td>Unconnected</td> <td>1970/1/1 0:00:00</td> </tr> <tr> <td>Connected</td> <td>Time when the WDT signal light information is received</td> </tr> <tr> <td>Disconnected</td> <td>Time when the WDT was determined to be disconnected</td> </tr> </tbody> </table>	State	TimeStamp variables	Unconnected	1970/1/1 0:00:00	Connected	Time when the WDT signal light information is received	Disconnected	Time when the WDT was determined to be disconnected								
State	TimeStamp variables																	
Unconnected	1970/1/1 0:00:00																	
Connected	Time when the WDT signal light information is received																	
Disconnected	Time when the WDT was determined to be disconnected																	

\* When the WDT is set to the request transmission mode, if get\_Value is executed between the state transition during the WDT connection and the first information request, it indicates "0, 0, 0, 0, 0, 0, 0, 0" (Fig. 4-9).



**Fig. 4-9 Request transmission mode get\_Value Result**

**Examples of Use (CaoScript)**

Dbg.output DAT.BstrFromVariant(varSignal.Value)

- '⇒Display result example:-1, -1, -1, -1, -1, -1,-1 (Unconnected)
- ' 2, 1, 4, 1, 0, 0, 98 (Connected)
- '-2,-2,-2,-2,-2,-2,-2,-2 (Disconnected)

Dbg.output DAT.BstrFromVariant(varTimeStamp.Value)

'⇒Display result example: 1/1/2019 0:00:00

### 4.3. Error-code

This provider defines a unique error code. The unique error code is shown in. Table 4-2 For ORiN2 common error, see the error code section in the ORiN2 Programming Guide.

**Table 4-2 Unique Error Codes**

Error Number	Description
0x80110000	The optional string Conn is incorrectly specified.
0x80120000	The optional string Interval is incorrectly specified. Check that the value is within the appropriate range.
0x80130000	The optional string FailureCount is incorrectly specified. Check that the value is within the appropriate range.
0x801400xx	Failed to start the network. For more information, see Wireless Data Communication System Application Notes 0x2D00 Network-Initiation Notification Status, which PATLITE the low-order two-digit number.
0x80150000	WDR was not received correctly when the WDT participation request command was sent.
0x80160000	Incorrectly received packets are present. Check the received packets by WireShark or other means.

Table 4-3 shows the error codes that commonly occur when using WDR providers.

**Table 4-3 Error codes that occur frequently**

Error Number	Description
0x8091274D	Error code for Windows sockets. Other software may be connected to the WDR. Disconnect the target software and WDR and try to connect again. Or, the port may be incorrect. Check the destination port number.

## 5. Sample code(CaoScript)

Sub Main

'Create controller

Dim ctrl

```
Set ctrl = cao.AddController("wdr", "CaoProv.PATLITE.WDR", "",  
                            "Conn=Eth:192.168.0.1:10001,Interval=1000, FailureCount = 3")
```

'Wait until a WDT is connected

Dim counter

counter = 0

Dim varNamesList

Do

    Ext.Delay(1000)

    counter = counter+1

    dbg.Output counter & "s"

    varNamesList = dat.ToVar(ctrl.VariableNames)

    If IsEmpty(varNamesList) = false Then

        Exit Do

    End if

Loop

'Create variables

Dim signalVar

Set signalVar = ctrl.AddVariable("743a65fffe4a0d79", "")

Dim timeStampVar

Set timeStampVar = ctrl.AddVariable("TimeStamp\_743a65fffe4a0d79", "")

Ext.Delay(2000)

'Output value

dim signalOutputVal

signalOutputVal = Dat.ChangeType(signalVar.value , 12) 'CaoScript can use only variant type

dbg.Output "Signal:" & Join(signalOutputVal)

dbg.Output "TimeStamp:" & timeStampVar.value

**Delete variables & controller**

Call `cao.Controllers("wdr").Variables.Remove(0)`

Set `signalVariable = Nothing`

Call `cao.Controllers("wdr").Variables.Remove(1)`

Set `timeStampSignalVar = Nothing`

**Delete controller**

Call `cao.Controllers.Remove(0)`

Set `ctrl = Nothing`

End Sub