Ishida Co., Ltd.
DACS-G providers

Version 1.0.0

User's Guide

December 1, 2020

NOTE:

[Revision History]

Version	Date	Description
1.0.0	2020-12-01	First edition

[Compatible devices]

Model	Version	Notes
DACS-G-S015		
DACS-G-S060		
DACS-G-S150		
DACS-G-S300		
DACS-G-S600		
DACS-G-F006		
DACS-G-F030		

[Operation check device]

Model	Version	Notes
DACS-G-S060	1.09J	

Table of Contents

1. Introduction

This manual is a user's guide for CAO providers that loads data into a weight checker (DACS-G) manufactured by Ishida, Inc.

The CAO provider (CaoProvIshidaDACS-G.dll) covered in this document is called DACS-G provider. DACS-G provider is developed in accordance with the "DACS-G Host Linkage Specification" provided by Ishida Co., Ltd. This document describes the features of DACS-G providers and the methods they implement.

2. Provider Overview

2.1. Introduction

DACS-G providers have a DLL (Dynamic Link Library) file format that is loaded dynamically when used by the CAO engine. To use DACS-G providers, you must install ORiN2 SDKs or manually register the registry by referring to Table 2-1.

Table 2-1 DACS-G Providers

File name	CaoProvIshidaDACS-G.dll
ProgID	CaoProv.Ishida.DACS-G
Registry register	Regsvr32 CaoProvIshidaDACS-G.dll
Deletion of Registry Registration	Regsvr32 /u CaoProvIshidaDACS-G.dll

2.2. Method properties

2.2.1. CaoWorkspace::AddController method

DACS-G providers refer to the connection parameters for communication during CaoWorkspace class-based AddController and make communication (TCP) connections. The specifi cations of AddController are shown below.

SYNOPSIS AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,<bstrPCName:BSTR>, <bstrOption:BSTR>)

BstrCtrlName	:	[in]	Controller name
BstrProvName	:	[in]	Provider name Fixed value = " CaoProv. Ishida.DACS-G"
BstrPCName	:	[in]	Provider's running machine name
BstrOption	:	[in]	Option character string

The following is a listing for Option character string:

Table 2-2 CaoWorkspace::AddController

Option	Description
Conn = <connection parameter>	Required. Communication format and connection parameters. (See2.2.1.1)
ConnTimeout[=<connection timeout>]	Specifies the timeout (in milliseconds) for the connection. Value range: 0-4294967295 (4294967295: Infinite wait) Default: 500
Timeout[=<timeout>]	Specifies the timeout (in milliseconds). Value range: 0-4294967295 (4294967295: Infinite wait) Default: 3000

2.2.1.1. Conn Optional

The following is a Conn optional connection parameter string: Here, brackets ("[]") are optional. In the explanation of each parameter, the underlined part becomes the default value when the option is not specified.

Ethernet Devices

"Conn=ETH:<Dest IP Address>[:<Dest Port No>[:<Src IP Address>[:Src Port No]]]"

"Conn=TCP: <Dest IP Address>[:<Dest Port No>[:<Src IP Address>[:Src Port No]]]"

<Dest IP Address>	:	Host address. Example: "127.0.0.1", "192.168.0.21"
<Dest Port No>	:	Number of the specific port 2026,5006,5007,... Optional
<Src IP Address>	:	Source IP address. (Multiple NIC Applications) Applications To automatically determine the IP address, specify "255.255.255.255". Examples: "127.0.0.1", "192.168.0.1", "255.255.255.255"
<Src Port No>	:	Source port number. (Multiple NIC Applications) Example: 0,5006,5007,... Can be specified arbitrarily

Usage

example

```
Dim caoCtrl As CaoController
Set caoCtrl = caoWorkspace.AddController("DACS-G", _
    "CaoProv.Ishida.DACS-G", "", _
    "Conn=TCP:192.168.0.21:2026,ConnTimeout=500,Timeout=3000")
```

2.2.2. CaoController::AddVariable method

AddVariable method of CaoController class is used to create a variable object that reads data from DACS-G.
For details about the variable names that can be specified, see 2.3Variable list

SYNOPSIS AddVariable(<bstrVariableName:VT_BSTR>[,<bstrOption:VT_BSTR>])

BstrVariableName	:	[in]	Variable name
BstrOption	:	[in]	Option character string

Usage

example

```
Dim varROMNumber As CaoVariable
Set varROMNumber = caoCtrl.AddVariable("@ROMNumber", "")
```

2.2.3. CaoController::property get_VariableNames

VariableNames property of CaoController class gets a list of system variables that can be specified in AddVariable method.

Refer to 2.3 Variable List for the variable names to be acquired.2.3Variable list

2.2.4. CaoController::Execute method

Execute method of CaoController class is a method for executing commands.

For details of each command, refer to 3 Command Reference.3Command Reference

SYNOPSIS Execute(<bstrCommandName:VT_BSTR>,[<vntParam:VT_VARIANT>])

BstrCommandName	:	[in]	Command name
VntParam	:	[in]	Parameter

Usage

example

```
Dim result As Variant
Result = caoCtrl.Execute("GetROMNumber")
```

2.2.5. CaoVariable::get_Value Property

Value property of CaoVariable class gets the value of the variable corresponding to the object.

For details of each variable, refer to 2.3Variable list

2.3. Variable list

2.3.1. Controller class

The following table lists the variables that can be used with CaoController.

2.3.1.1. System Variables

The following table lists the system variables that can be used with CaoController.

Table 2-3: List of controller class system variables

Variable name	Data Type	Description	Attribute	
			Get	Put

@MAKER_NAME	VT_BSTR	Returns the manufacturer named "Ishida".	✓	-
@VERSION	VT_BSTR	Returns the version of the provider DLL.	✓	-
@ROMNumber	VT_ARRAY | VT_VARIANT	Returns the ROM number.	✓	-
@Status	VT_ARRAY | VT_VARIANT	Returns the state.	✓	-
@TotalsParameters	VT_ARRAY | VT_VARIANT	Returns the aggregation condition.	✓	-
@RejectFunctionSelection	VT_ARRAY | VT_VARIANT	Returns the sort function selection.	✓	-
@WeightHistory	VT_ARRAY | VT_VARIANT	Returns the weighing history.	✓	-

2.3.1.1.1. @MAKER_NAME

Returns the manufacturer named "Ishida".

VT_BSTR	Manufacturer name "Ishida"

2.3.1.1.2. @VERSION

Returns the version information of the provider DLL.

VT_BSTR	Provider DLL version information

2.3.1.1.3. @ROMNumber

Returns the ROM number.

VT_ARRAY | VT_VARIANT			
	0	VT_BSTR	ROM number and version "0000" – "9999Z"
	1	VT_BSTR	System version number (upper) "01" – "99"
	2	VT_BSTR	System version number (low) "01" – "99"
	3	VT_BSTR	Model code "0" : ACP

			"1" : CCW-S "2" : MCW-2XXX "3" : LCW-11X "4" : CCW-CD "5" : CCW-RLC "6" : CCW-Z "7" : DACS-V "8" : CCW-RZ "9" : CCW-DZ ":" : DACS-W & WN ";" : DACS-H "<" : DACS-G
	4	VT_UI1	Whether or not DACS-G is optional Bit 0 : Average control Bit 1 : Bit 2 : Bit 3 :
	5	VT_UI1	e/10 indication 0 : None 1 : With
	6	VT_BSTR	DACS-G model "0" : S015 "1" : S060 "2" : S150 "3" : S300 "4" : S600 "5" : F006 "6" : F030 "7" : "8" : "9" :
	7	VT_UI1	Applicable Countries 0 : Japan 1 : 2 : 3 :

			4 : U.S.A. 5 : 6 : 7 : 8 : 9 : 10 : Others
	8	VT_BSTR	Host-linked (Weight-Net) communication spec. version number "0.00 " - "9.99Z"

2.3.1.1.4. @Status

Returns the operating status.

VT_ARRAY | VT_VARIANT			
	0	VT_BSTR	Reservation number "0000" – "0200"
	1	VT_UI1	DACS-G status "0": Stopping "1" : Running "3": Error stopped (Error screen is displayed.)
	2	VT_UI1	Online 0 : Offline 1 : Online
	3	VT_BSTR	Error code "00" : No error "01" : Zero point error "02": Load conveyer error (overload) "03": Capture synchronous conveyor error (overload) "04": Weighing conveyor error (overload) "05": Distribution conveyer error (overload) "06": Distribution synchronous conveyer error (overload) "07": Extra conveyer error (overload) "08": Import conveyer error (pulse error) "09": Weighing conveyor error (pulse error) "10": Distribution conveyer error (pulse error)

			"11" : Additional conveyor error (pulse error) "12" : ADC error "14" : Right sort error "15" : Left sort error "18" : Transport failure (medium, Iffs) "21" : Transport error (in progress) "22" : Transport error (right) "23" : Transport error (left) "25" : Photoelectric sensor error "28": Air pressure drop error "35" : Setting error "99" : Not applicable

2.3.1.1.5. @TotalsParameters

Returns the aggregation condition.

VT_ARRAY | VT_VARIANT			
	0	VT_UI1	Object of calculation 0 : Total number 1 : Shipments
	1	VT_UI1	Batch aggregation 0 : No 1 : Total number 2 : Time 3 : Shipments
	2	VT_UI1	Automatic counting and printing 0 : No 1 : To do
	3	VT_UI1	Batch aggregate printing 0-23 hours
	4	VT_I4	Batch Count

2.3.1.1.6. @RejectFunctionSelection

Returns the sort function selection.

VT_ARRAY | VT_VARIANT

	0	VT_UI1	Select sort function 0 : Batch 1 : Each reservation

2.3.1.1.7. @WeightHistory

Returns the weighing history.

VT_ARRAY | VT_VARIANT							
	0	VT_UI1	Number of data 0 - 16				
	1	If the number of data is 0, VT_EMPTY	Results				
		If the number of data is 1 or more, VT_ARRAY | VT_VARIANT					
			0	VT_ARRAY | VT_VARIANT	Result 1		
				0	VT_BSTR	Sort Result 1 "01" : Positive amount ● "02" : Light weight ○ "03": Overdose △ "04" : Mixed metal "05" : External 1 "06" : External 2 "07" : Ride 2 "08" : Zero point error "09": Continuous failure "10" : Missing metal "11": Positive light weight ● "12" : Too much ● "13" : Light failure "14" : Bad "15" : Foreign object error "16" : Short Length "17" : Double bag "18" : External 3 "19" : External 4 <Totals>

							All → ● ○ Weighing items marked with a △ Shipped products (overquantity shipments: shall be made) → ● Items marked with a △ Shipped goods (overquantity shipment: not shipped) → ● Weighing items marked with
					1	VT_R8	Weighing data 1
					2	VT_BSTR	Serial number 1 "00000001" - "99999999"
			1	VT_ARRAY | VT_VARIANT			Result 2
					0	VT_BSTR	Sort Result 2
					1	VT_R8	Weighing data 2
					2	VT_BSTR	Serial number 2
			・・・				

2.3.1.2. User variable

There are no user variables available in the CaoController class.

3. Command Reference

3.1. Controller classes

List of 3-1 CaoController::Execute Commands

3.1.1. CaoController::Execute("GetROMNumber") Command

Gets the ROM number.

SYNOPSIS GetROMNumber()

Argument : None
Return Value : ROM number (VT_ARRAY | VT_VARIANT)
@ROMNumber

Usage

example

```
Dim result As Variant
Result = caoCtrl.Execute("GetROMNumber")
```

3.1.2. CaoController::Execute("GetStatus") Command

Gets the operating status.

SYNOPSIS GetStatus()

Argument : None
Return Value : Status (VT_ARRAY | VT_VARIANT)
@Status

Usage

example

```
Dim result As Variant
Result = caoCtrl.Execute("GetStatus")
```

3.1.3. CaoController::Execute("GetTotalsParameters") Command

Gets the aggregation condition.

SYNOPSIS GetTotalsParameters()

Argument	:	None
Return Value	:	Aggregation criteria (VT_ARRAY | VT_VARIANT) @TotalsParameters

Usage

example

```
Dim result As Variant
Result = caoCtrl.Execute("GetTotalsParameters")
```

3.1.4. CaoController::Execute("GetRejectFunctionSelection") Command

Obtain the sort function selection.

SYNOPSIS GetRejectFunctionSelection()

Argument	:	None
Return Value	:	Select sort function (VT_ARRAY | VT_VARIANT) @RejectFunctionSelection

Usage

example

```
Dim result As Variant
Result = caoCtrl.Execute("GetRejectFunctionSelection")
```

3.1.5. CaoController::Execute("GetWeightHistory") Command

Gets the weighing history.

SYNOPSIS GetWeightHistory ()

Argument	:	None
Return Value	:	Weighing History (VT_ARRAY | VT_VARIANT) @WeightHistory

Usage

example

```
Dim result As Variant
Result = caoCtrl.Execute("GetWeightHistory")
```

4. Error code

DACS-G providers define the following unique error codes:

For ORiN2 common error codes, refer to the Error Codes section of ORiN2 Programming Guide.

Table 4-1 Unique error codes

Error name	Error Number	Description
Response that cannot be processed	0x80100000	Returned if DACS-G cannot successfully process the request.
Abnormal response	0x80100001	Returned if DACS-G received the following: １. Undefined command received ２. Sum check error ３. Receive text data length error ４. Incorrect characters in received text
Receive data error (Data length)	0x80110000	Returned when the number of received data items from DACS-G is abnormal.
Receive data error (no line feed code)	0x80110001	Returned if there is no line feed code in the data received from DACS-G.
Receive data error (without escape code)	0x80110002	Returned when there is no escape code in the data received from DACS-G.
Receive data error (Command)	0x80110003	It is returned when the command of the data received from DACS-G is abnormal.
Receive data error (sum check)	0x80110004	Returned when the sum check of the data received from DACS-G is abnormal.

5. Communication protocol command correspondence table

Table 5-1 shows the correspondence between Execute and Variable methods implemented by this provider and the communication commands in DACS-G Host Linkage Specification Table 5-1.

Table 5-1 Correspondence between communication commands

Execute method	Variable name	Get/put	Communication Commands
GetROMNumber	@ROMNumber	Get	ROM number request Command: "," - > ROM number response Command: "Y"
GetStatus	@Status	Get	Operating status request Command: "@" -> Operational state response Command: "9"
GetTotalsParameters	@TotalsParameters	Get	Aggregation Requirements Command: """ -> Aggregation condition response Command: "T"
GetRejectFunctionSelection	@RejectFunctionSelection	Get	Sort function selection request Command: "L" -> Sort function selection response Command: "O"
GetWeightHistory	@WeightHistory	Get	Weighing history request Command: "H" -> Weighing history response Command: "H"

6. Sample program

Here is an example using DACS-G providers:

List 5-11 Sample.frm

```
Option Explicit

Private m_Engine As CaoEngine
Private m_Workspace As CaoWorkspace
Private m_Controller As CaoController
Private m_Variable As CaoVariable

Private Sub cmdExexute_Click()

    ' Execute "GetStatus"
    Dim result As Variant
    Result = m_Controller.Execute("GetStatus")
    TxtPresetNumber.Text = result(0)
    TxtDACSGMode.Text = CStr(result(1))
    TxtOnLine.Text = CStr(result(2))
    TxtErrorCode.Text = result(3)

End Sub

Private Sub cmdValue_Click()

    ' Get the variable value of "@Status"
    TxtPresetNumber.Text = m_Variable.Value(0)
    TxtDACSGMode.Text = CStr(m_Variable.Value(1))
    TxtOnLine.Text = CStr(m_Variable.Value(2))
    TxtErrorCode.Text = m_Variable.Value(3)

End Sub

Private Sub Form_Load()

    ' Connect
    Set m_Engine = New CaoEngine
    Set m_Workspace = m_Engine.Workspaces(0)
    Set m_Controller = m_Workspace.AddController("DACS-G", "CaoProv.Ishida.DACS-G", _
        "", "Conn=TCP:192.168.0.21:2026,ConnTimeout=500,Timeout=3000")
    ' Add CaoVariable
    Set m_Variable = m_Controller.AddVariable("@Status")

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

    ' Disconnect
    m_Controller.Variables.Remove m_Variable.Index
    Set m_Variable = Nothing
    m_Workspace.Controllers.Remove m_Controller.Index
    Set m_Controller = Nothing
    Set m_Workspace = Nothing
    Set m_Engine = Nothing

End Sub
```