VBP provider

Gateway for VB6 provider

Version 1.2.0

User's guide

July 17, 2012

【 remarks 】

【 revision history 】

Version	Date	Content
1.0.0.0	2007-04-06	First edition.
1.0.1.0	209-09-07	VB provider making procedure and programming Tips postscript.
1.0.1.1	2010-02-12	Addition of error code.
1.1.1.0	2010-09-16	Addition of method of displaying modeless dialog.
1.2.0.0	2011-12-26	Improved FinalInitialize function of Caop4vbMessage.
1.2.0	2012-07-17	Document versioning rules was changed.

【 hardware 】

Model	Version	Notes

【 attention 】

When this provider is used, the “ORiN2 VBGateway Provider” license is separately necessary.

Contents

1. Introduction

This document is an user's guide of the Gateway provider (henceforth VB_Gateway provider) for the VB provider.

As for the current provider development, advanced knowledge to C++, COM (Component Object Model), and the multipurpose data type was not able to be developed necessary, and easily. It proposes the provider development by Visual Basic 6.0 that the more mounting is handy as one of these solution for the problems. As a result, developing more easily than the provider until the past becomes possible, and enhancing it becomes possible also the kind of the provider further.

The VB_Gateway provider plays the role as the gateway provider that wraps VB Provider and converts it into the CAO Provider specification.

2. Outline of provider

2.1. Outline

The interface specification of the VB provider is a specification near the interface specification of the CAO provider. However, it is difficult to reproduce the same interface by restricting Visual Basic. Then, the VB_Gateway provider absorbs this difference by wrapping the VB provider and offers the same function as the CAO provider.

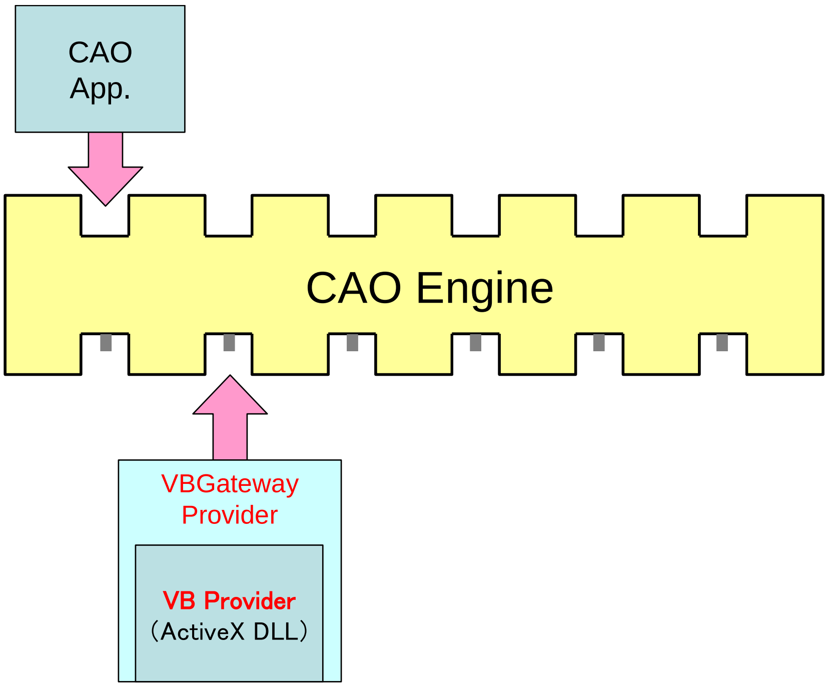


Figure1-1Architecture

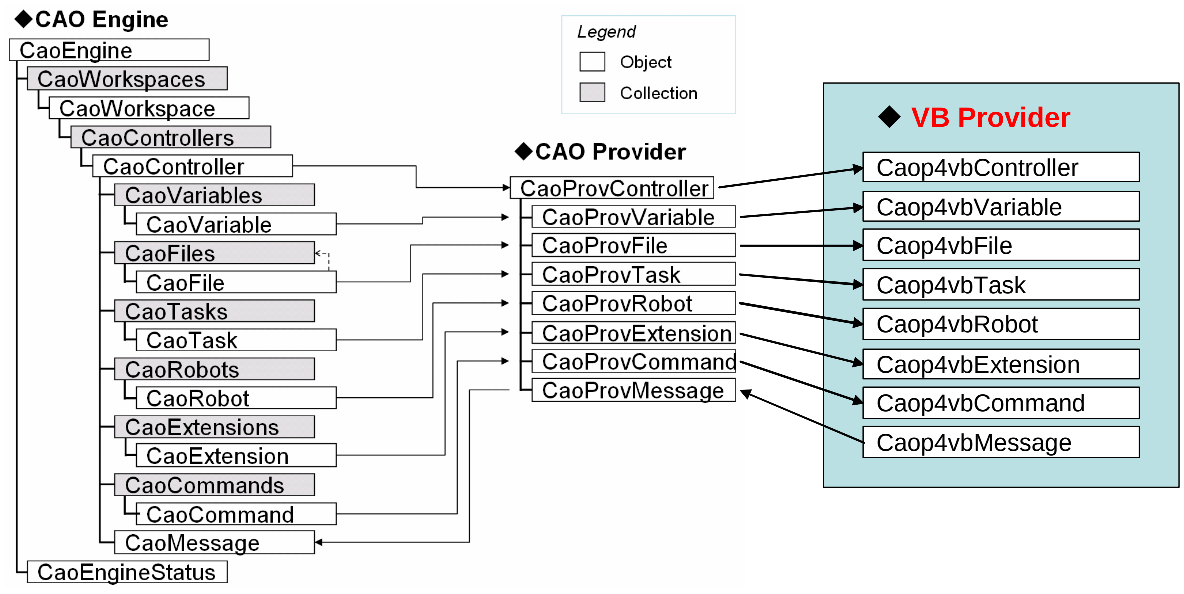


Table1-1VB_Gateway provider

File name	CaoProvVBP.dll
ProgID	CaoProv.VBGateway
Registry registration	regsvr32 CaoProvVBP.dll
Blotting out of registry registration	regsvr32 /u CaoProvVBP.dll

2.2. Method property

2.2.1. CaoWorkspace::AddController method

It connects it with the VB provider in the VB_Gateway provider at AddController.

At this time, the VB provider that uses it by the option is specified with ProgID.

Format AddController (< bstrCtrlName:BSTR >, < bstrProvName:BSTR >
<bstrPcName:BSTR > [,<bstrOption:BSTR>])

bstrCtrlName	: In controller name
bstrProvName	: In provider name. " CaoProv.VBGateway fixed value ="
bstrPcName	: Execution machine name of in provider
bstrOption	: In option character string

The list specified for the option character string is shown as follows.

Table 2-1 Option character string of CaoWorkspace::AddController

Option	Explanation
VBP=< ProgID>	ProgID of VB provider It agrees to the project name of the VB provider.

2.2.2. Method properties other than AddController

The controller, the robot, the file, the task, the variable, and all the methods and the properties of the extension board class are mounted as for the VB Gateway provider, and they call the function that the VB provider responds as it is.

2.3. Error code

In the VB Gateway provider, there is no peculiar error code. Please refer to the chapter of the error code of "ORiN2 Programming guide" for the ORiN2 commonness error.

3. Sample program

List 3-1 **Sample.frm**

```
Private Eng As CaoEngine
Private Ctrl As CaoController
Private Var As CaoVariable

Private Sub Form_Load()

 ..'.. generation of CAO engine
 Set Eng = New CaoEngine

The controller's ..'.. connection
 Set Ctrl = Eng.Workspaces(0).AddController("Sample", "CaoProv.VBGateway", "",
"VBP=CaoProv_Sample_Macro")

 ..'.. call of method of VB provider
 Set Var = Ctrl.AddVariable("Var", "")
 MsgBox Var.Value

End Sub
```

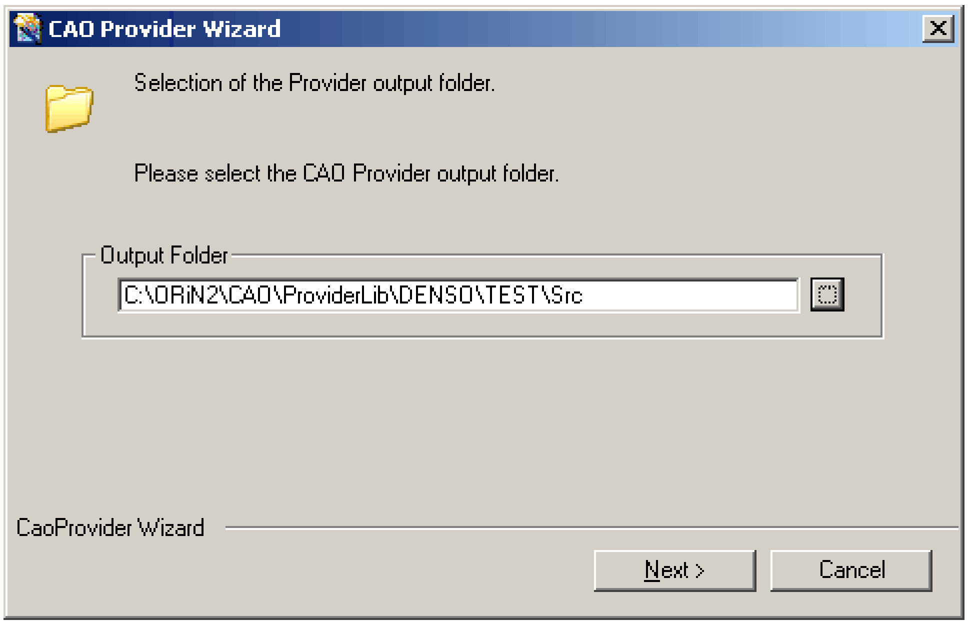
4. VB provider making procedure

4.1. Making of skeleton

The skeleton project of Visual Basic6.0 is made by using the CAO provider wizard.

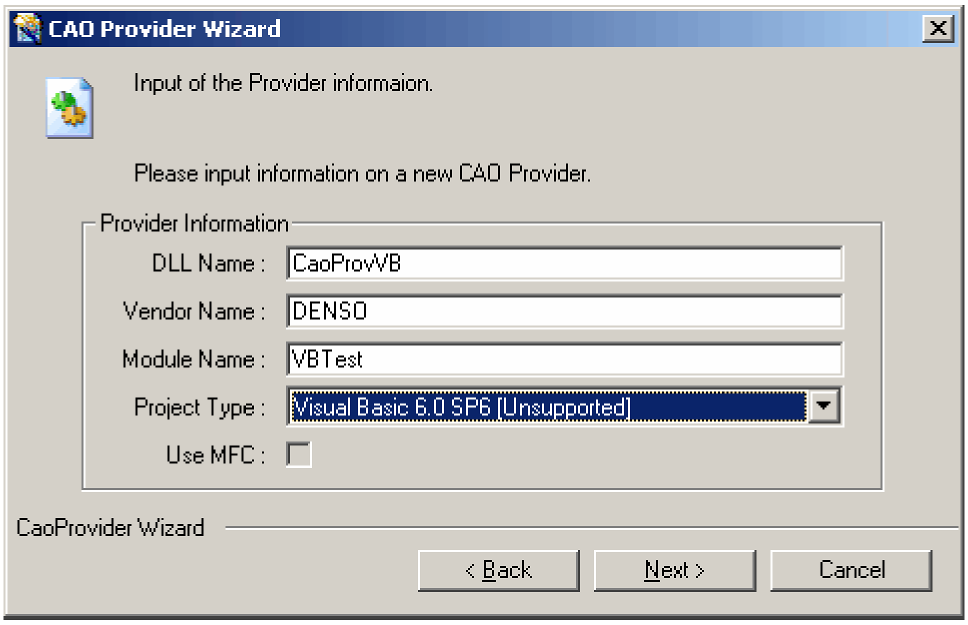
(1) "All programs" →"ORiN2" →"CAO" →"Provider" →"CaoProvWizard" of the start menu is started.

(2) Passing at the installation destination is specified for Output Folder.


CAO Provider Wizard
Selection of the Provider output folder.
Please select the CAO Provider output folder.
Output Folder
C:\ORiN2\CAO\ProviderLib\DENSO\TEST\Src
CaoProvider Wizard
Next >
Cancel

Specification of Figure 4–1 passing

(3) "Visual Basic 6.0 SP6" is selected, and DLL Name, Vendor Name, and Module Name are input to Project Type.


CAO Provider Wizard

Input of the Provider informaion.

Please input information on a new CAO Provider.

Provider Information

DLL Name : CaoProvVB

Vendor Name : DENSO

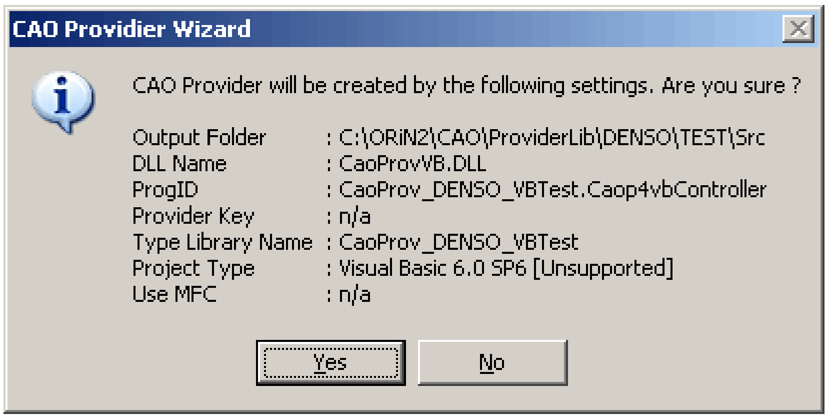
Module Name : VBTest

Project Type : Visual Basic 6.0 SP6 [Unsupported]

Use MFC : ☐

Setting of Figure 4-2 provider

(4) Yes (Y) is pushed ..confirmation screen.. [demasunode].

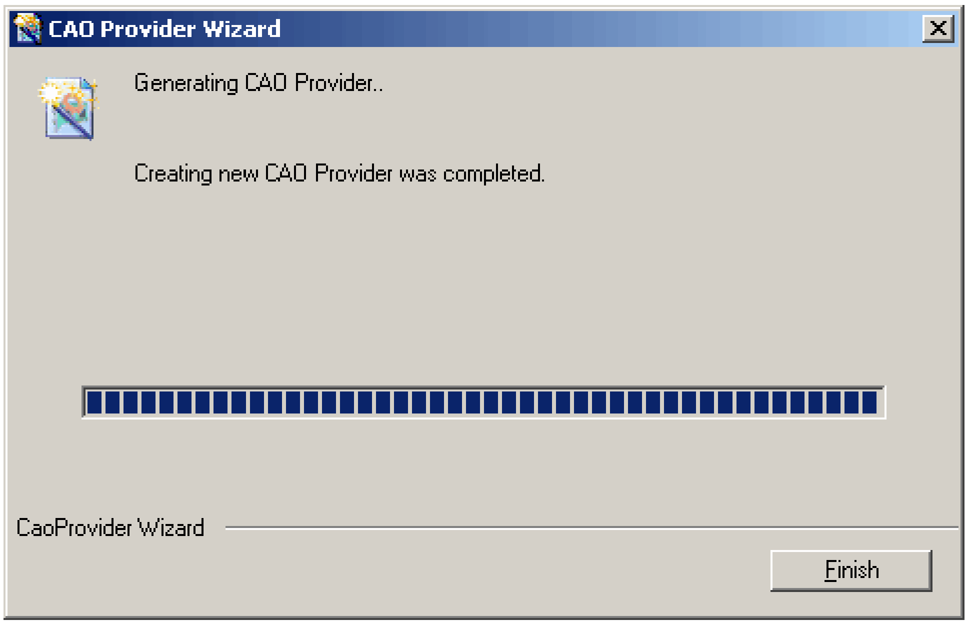

CAO Providier Wizard

CAO Provider will be created by the following settings. Are you sure ?

```
Output Folder      : C:\ORiN2\CAO\ProviderLib\DENSO\TEST\Src
DLL Name           : CaoProvVB.DLL
ProgID             : CaoProv_DENSO_VBTest.Caop4vbController
Provider Key       : n/a
Type Library Name  : CaoProv_DENSO_VBTest
Project Type       : Visual Basic 6.0 SP6 [Unsupported]
Use MFC            : n/a
```

Set confirmation of Figure 4-3 provider

(5) If the following screen goes out, it is making completion.


CAO Provider Wizard

Generating CAO Provider..

Creating new CAO Provider was completed.

CaoProvider Wizard

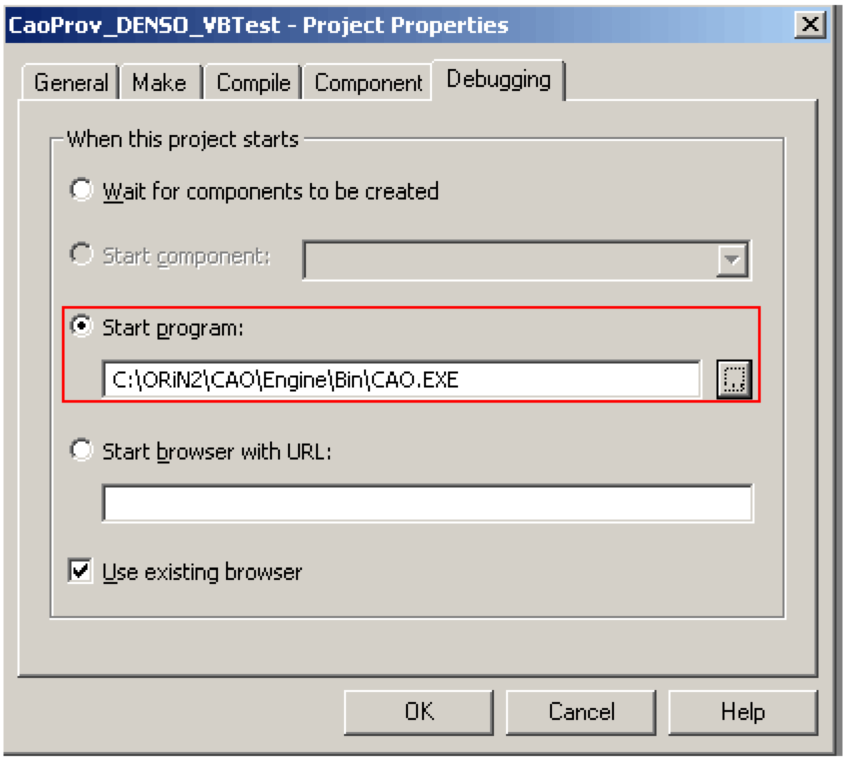
Figure 4-4 skeleton making completion

4.2. Mounting of function

Because CaoProv.vbp is made for the folder specified ahead, a necessary function is mounted.

4.3. Debugging

CAO engine (Cao.exe) is specified in the debugging tab of the project property.


CaoProv_DENSO_VBTest - Project Properties

General | Make | Compile | Component | Debugging

When this project starts

( ) Wait for components to be created

( ) Start component:

(●) Start program:

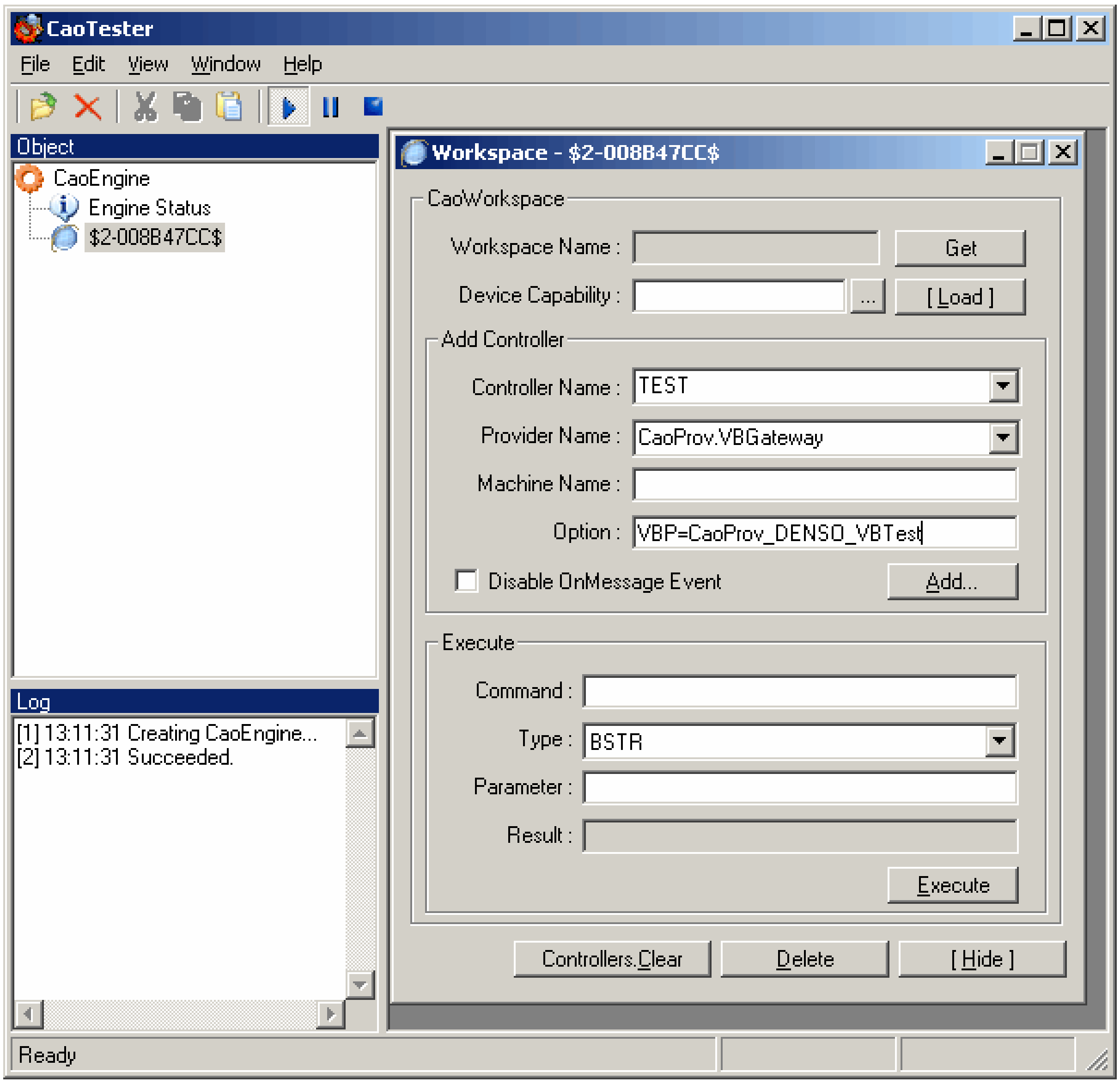
C:\ORiN2\CAO\Engine\Bin\CAO.EXE

( ) Start browser with URL:

[✓] Use existing browser

Figure 4-4 Specification of CAO engine

Debugging begins, and a suitable client program (The example: CaoTester) is started.



Controller Name : TEST
Provider Name : CaoProv.VBGateway
Option : VBP=CaoProv_DENSO_VBTest
Type : BSTR

```
[1] 13:11:31 Creating CaoEngine...
[2] 13:11:31 Succeeded.
```

Example of testing Figure 4-4 debugging

5. Programming Tips

5.1. OnMessage event

5.1.1. Procedure

To generate the OnMessage event, the following procedures are done.

(1) The timer flag is turned on with Caop4vbController_Initialize().

(2) Message object (Caop4vbMessage) is generated with the CreateMessage function of the Caop4vbConttoller class.

(3) The Caop4vbMessage class is mounted. Please note that a value whether a pVal argument of FinalInitialize is VT_EMPTY or not specifies whether it called from VBGateway Provider (external) or a Caop4vbController Object (internal).

5.1.2. Sample

List 5-1-1 **Caop4vbController**

```
Private Sub ICaop4vbController_FinalConnect(pVal As Variant)

    ‘ Set timer flag to ON.
    pVal(2) = True

    ‘ Create a message object
    CreateMessage  “TestMsg”
End Sub
```

List 5-1-2 **Caop4vbMessage**

```
Private m_vntData As Variant

‘ Initialize a message object.
Private Sub ICaop4vbMessage_FinalInitialize(pVal As Variant)
    If IsEmpty(pVal) Then
        ‘ Called from VBGateway Provider
    Else
        ‘ Called from Caop4vbController
        m_vntData = pVal
    End If
End Sub

‘ Get value
Private Function ICaop4vbMessage_FinalGetValue() As Variant
    ICaop4vbMessage_FinalGetValue = m_vntData
End Function
```

5.2. Parents object information

5.2.1. Procedure

To acquire information intended for parents with FinalInitialize, the following procedures are done.

(1) The public member is added to the parents object.

(2) The public member is called from the parents object that exists in the argument of FinalInitialize().

5.2.2. Sample

List 5-2-1 Caop4vbController (parents object)

```
Option Explicit
Implements CAOPROV4VB.ICaop4vbController

Public TempData As String

Private Sub ICaop4vbController_FinalConnect(pVal As Variant)
    TempData = "This is VB Test!!"
End Sub
```

List 5-2-2 Caop4vbVariable (child object)

```
Private Sub ICaop4vbVariable_FinalInitialize(pVal As Variant)

    Dim Ctrl As Caop4vbController
    Set Ctrl = pVal(1)

    OutputDebugString Ctrl.TempData

End Sub
```

5.3. Micro provider

5.3.1. Procedure

How to make the micro provider is shown below.

(1) The CAO provider that uses it is added from the reference configuration of the project.

(2) The object of the CAO provider is declared.

(3) Each method is mounted.

5.3.2. Sample

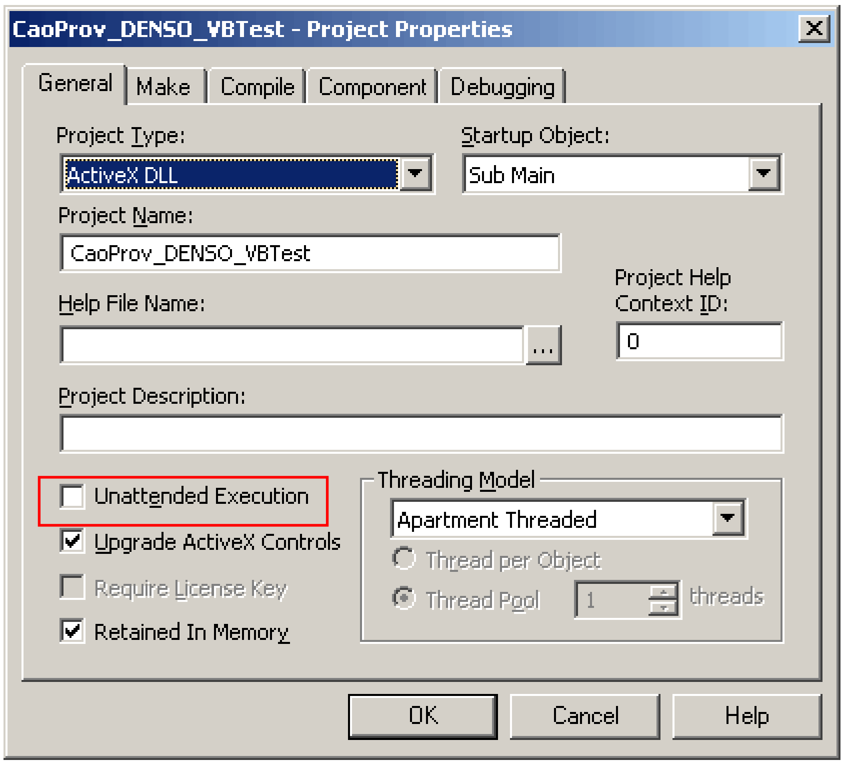
List 5-3

```
Implements CaoProv4VB.ICaop4vbController

 ..'.. declaration of CAO provider object.
Private m_queMsg As New Collection
Public WithEvents m_ProvCtrl As CaoProvController

 Generation and ..'.. initialization of CAO provider object.
Private Sub ICaop4vbController_FinalConnect(pVal As Variant)
 ' Connect Controller
 Set m_ProvCtrl = New CaoProvController
 m_ProvCtrl.Connect "Test", "Option=Sample"

 ' Timer on
 pVal(2) = True
End Sub

 ..'.. disconnection process of CAO provider object.
Private Sub ICaop4vbController_FinalDisconnect()
 m_ProvCtrl.Disconnect
End Sub

 ..'.. call of method of CAO provider.
Private Function ICaop4vbController_FinalGetHelp() As String
 ICaop4vbController_FinalGetHelp = m_ProvCtrl.Help
End Function

 ..'.. message processing from CAO provider.
 The message object of the VB provider is made, and the message of the CAO provider ..'.. is stored.
Private Sub m_ProvCtrl_OnMessage(ByVal pCaoProvMessage As _
                                 CAOPROVLib.ICaoProvMessage, _
                                 Optional ByVal lOption As Long = 0&)
 CreateMessage pCaoProvMessage
End Sub
```

5.4. Display of modal dialog

5.4.1. Procedure

(4) Turn off [Unattended execution] option in the project property window.


CaoProv_DENSO_VBTest - Project Properties

General | Make | Compile | Component | Debugging

Project Type: ActiveX DLL

Startup Object: Sub Main

Project Name: CaoProv_DENSO_VBTest

Help File Name:

Project Help Context ID: 0

Project Description:

☐ Unattended Execution

☑ Upgrade ActiveX Controls

☐ Require License Key

☑ Retained In Memory

Threading Model: Apartment Threaded

○ Thread per Object

● Thread Pool 1 threads

(5) Create a form (dialogue).

(6) Show the form with vbModal option.

5.4.2. Sample

List 5-4

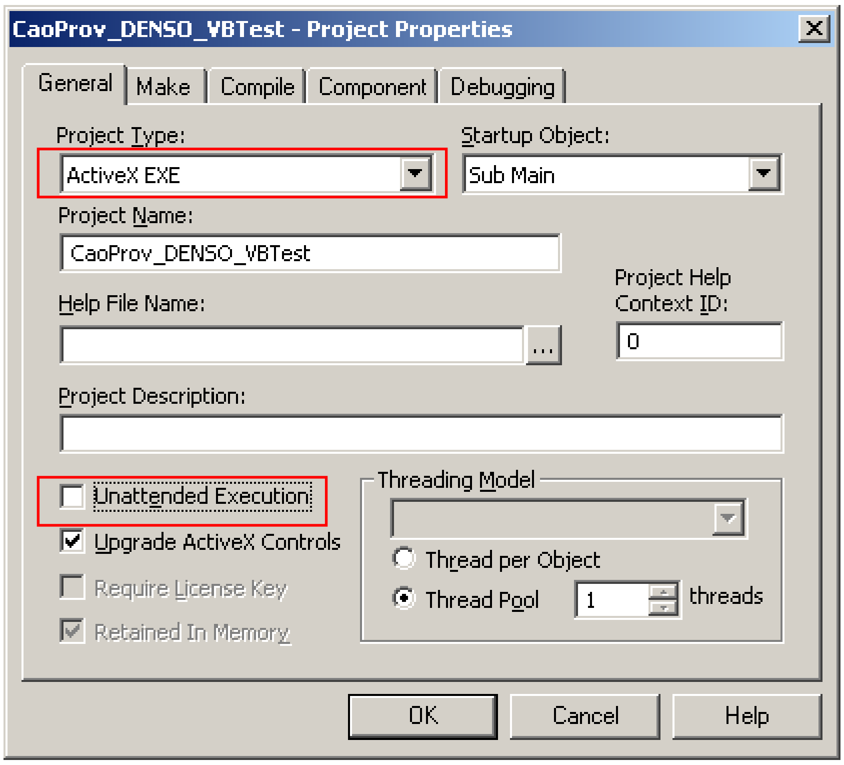
```
Private Sub ICaop4vbController_FinalConnect(pVal As Variant)

 ' Set initial parameters
 Load frmConnect
 frmConnect.InitParam pVal

 ' Show a connection dialogue
 frmConnect.Show vbModal

 If (frmConnect.Connected = False) Then
 Err.Raise vbObjectError + 1, , "Connection Error!"
 End If

End Sub
```

5.5. Display of modeless dialog

5.5.1. Procedure

(1) Select "ActiveX EXE" type in "General" tab of the project property window.

(2) Turn off [Unattended execution] option.



CaoProv_DENSO_VBTest - Project Properties

General | Make | Compile | Component | Debugging

Project Type: ActiveX EXE

Startup Object: Sub Main

Project Name: CaoProv_DENSO_VBTest

Help File Name:

Project Help Context ID: 0

Project Description:

[ ] Unattended Execution

[✓] Upgrade ActiveX Controls

[ ] Require License Key

[✓] Retained In Memory

Threading Model

( ) Thread per Object

(●) Thread Pool 1 threads

(3) Create a form (dialogue).

(4) Show the form.

5.5.2. Sample

List 5-5

```
Private Sub ICaop4vbController_FinalConnect(pVal As Variant)

 ' Set initial parameters
 Load frmConnect
 frmConnect.InitParam pVal

 ' Show a connection dialogue
 frmConnect.Show

 If (frmConnect.Connected = False) Then
 Err.Raise vbObjectError + 1, , "Connection Error!"
 End If

End Sub
```