b-CAP provider

b-CAP communication

Version 1.4.1

User's guide

December 4, 2024

[remarks]

This document is translated from Japanese into English by the machine translation.

[Revision history]

Version	Date	Content
1.0.0.0	2006-08-02	First edition.
1.0.1.0	2007-06-23	Interval option was added
1.0.2.0	2007-11-21	UDP option was added.
1.0.3.0	2008-01-19	Explanation supplementation of bCapListener.
1.0.4.0	2009-04-01	MyIP option was added.
1.1.0.0	2009-07-21	Priority setting options (/R) was added to b-CAP Listener, UDP retry function was supported.
1.1.1.0	2010-02-10	Error code was added, bCapService and bCapConfig was added
1.1.2.0	2010-08-20	b-CAP/COM communication was supported.
1.1.3.0	2011-05-15	BcapConfig correction was modified.
1.1.4.0	2012-06-05	The maximum packet size of each communication mode was set.
1.1.4	2012-07-17	Document versioning rules were changed.
1.2.0	2012-08-06	WDT option and AsyncCancel option were added.
1.2.1	2012-09-06	KeepAlive time option and InvokeTimeout option were added.
1.2.2	2013-01-29	b-CAP/ZIP Compression was supported.
1.3.0	2014-10-08	Support secure communication by SSL
1.3.1	2016-04-13	Added the control of the maximum number of b-CAP/UDP connection.
1.3.2	2018-11-29	Support continuous connection.
1.3.3	2018-12-17	Added retry of handshake during SSL communication.
1.3.4	2020-11-01	Memory-related processing fixes. Overall process corrected.
1.3.5	2021-08-19	Fixed processing at startup and termination. Fixed reconnection processing. Fixed memory related processing. Fixed exclusive processing. Support for asynchronous message processing.
1.4.0	2021-12-20	OpenSSL version upgrade.
1.4.1	2022-01-06	Fixed a bug in SSL option.
	2022-01-31	Change of certificate creation method.
	2024-12-04	OpenSSL version upgrade.

[Operation check model]

Model	Version	Notes

Contents

1. Introduction

This book is a user's guide of the b-CAP provider which is a provider to communicate with CAO of the remote machine by the use of b-CAP.

b-CAP is a protocol that aims at the improvement of the transmission rate following the concept of CAP. Therefore, b-CAP offers the function similar to CAP by using the TCP stream communication.

This book explains the function of the b-CAP provider and the mounting method.

This product includes software developed by the OpenSSL Project[1] for use in the OpenSSL Toolkit.

[1] https://www.openssl.org/

2. Outline of provider

2.1. Outline

The b-CAP provider adopts b-CAP (Binary CAP) as a communication specification. b-CAP is a protocol that aims at the improvement of the transmission rate following the concept of CAP.

The comparison chart in other remote communication forms is shown as follows.

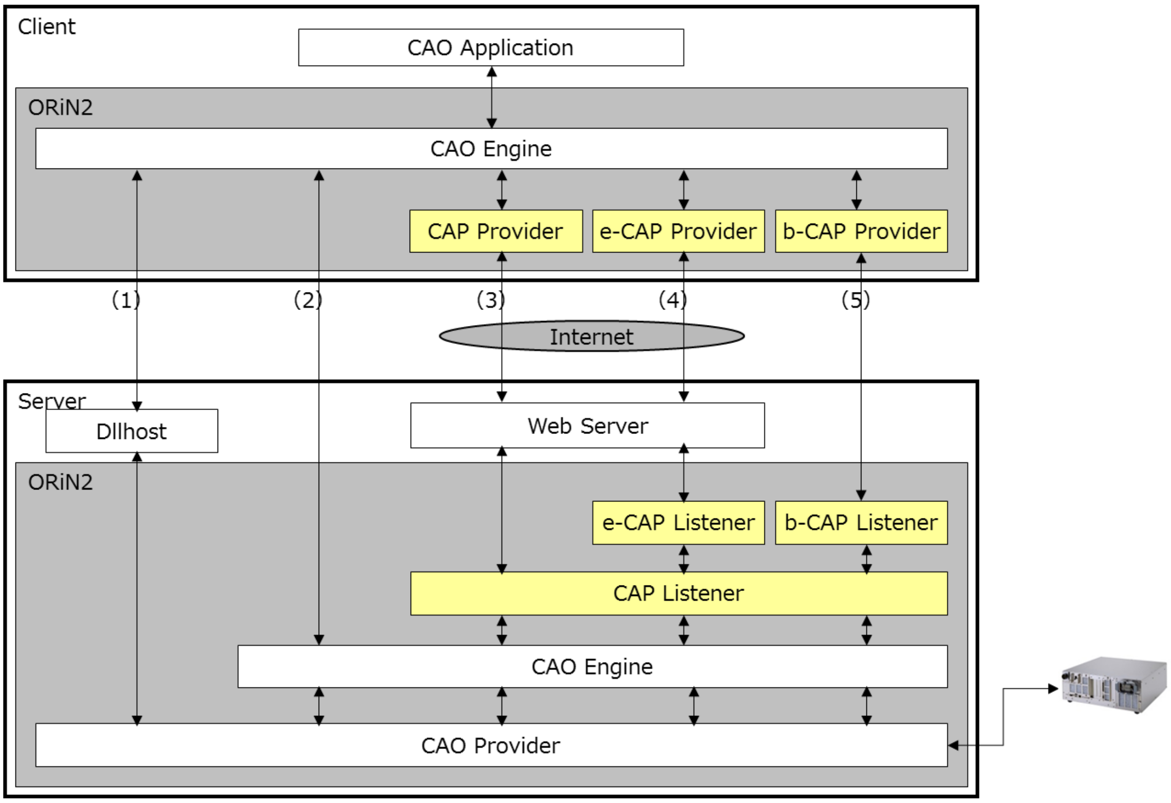


Figure 2-1 Comparison of communication forms

b-CAP communicates a b-CAP message that expresses the method call and the execution result by using TCP. Please refer to b-CAP specifications for details of the communication protocol of b-CAP.

There is b-CAP listener as a program to process the message transmitted from the b-CAP provider on the server side. The b-CAP listener executes the method of CAO specified by the message through the CAP listener.

b-CAP provider and b-CAP listener enable remote CAO engine operation by b-CAP.

The example of connecting b-CAP is shown as follows. Number [3] in Figure 2-2 shows the connecting example between b-CAP provider and b-CAP listener.

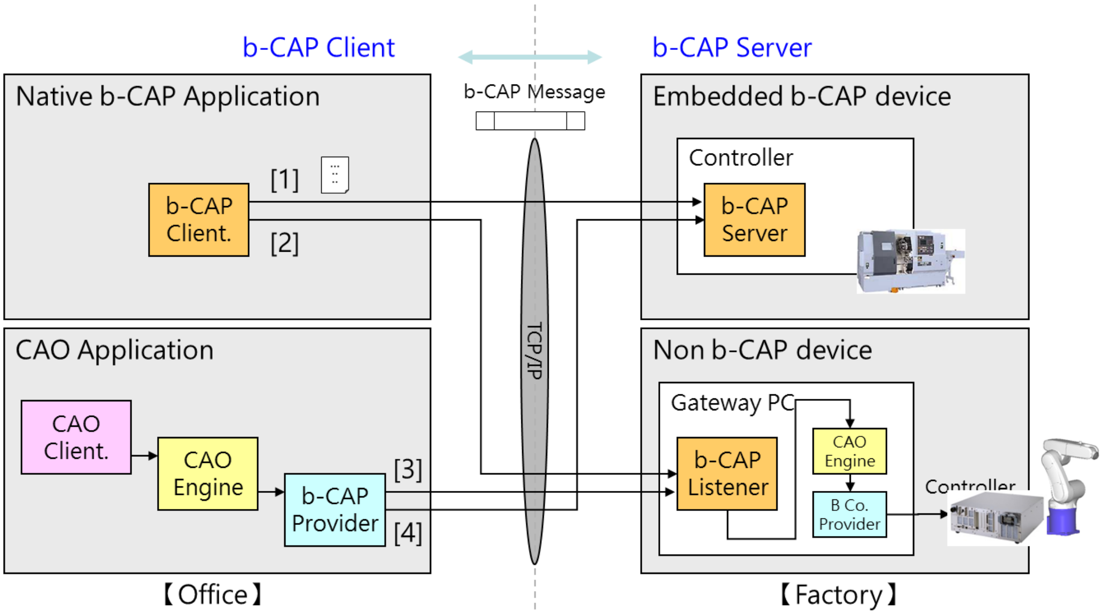


Figure 2-2 Connection status by b-CAP

In the b-CAP provider, three kinds (TCP, UDP, and the COM communication) are prepared as a communication method. Select a desirable communication method when connecting.

In the b-CAP provider, the maximum size of the packet is different depending on the communication method.

Table2-1 Size of the maximum packet

Communication method	Size of the maximum packet
TCP	4G Bytes
UDP	504 Bytes
COM	504 Bytes

2.1.1. Setup of b-CAP provider

To use the b-CAP provider, it is necessary to register in the registry to refer from CAO.

Table2-2 b-CAP provider

File name	CaoProvBCAP.dll
ProgID	CaoProv. b-CAP
Registry registration[2]	regsvr32 CaoProvBCAP.dll
Deregistration	regsvr32 /u CaoProvBCAP.dll

2.1.2. Message

The acquisition of the message created by a remote engine can be switched by the “Message" option of the AddController method.

When the message is turned off, the message created by a remote engine cannot be acquired.

When the message is turned on, the message that the bit flag of the engine control message in the message option is OFF can only be acquired..

Moreover, the message cannot be acquired at times in the execution of other methods. The message created while executing method is acquired collectively after current message execution.

2.1.3. Secure communication by SSL

For b-CAP Version 1.3.0 or higher, you can use SSL for secure communication.[3]

SSL achieves secure communication by encrypting/decrypting existing TCP communication. To use SSL for secure communication, you need to properly set up both the b-CAP provider and the program on the server-side (such as b-CAP listener).

For details about setup, refer to “6.1.Setup Procedure for Secure Communication by SSL”.

.

.

[2] It is not necessary manual registration/deregistration when installing it with ORiN SDK.

[3] Secure communication using SSL is only applicable to TCP, not UDP.

2.2. Method and property

2.2.1. CaoWorkspace::AddController method

When this method is executed, CAO and the CAO provider are started and connected in the server. At this time, this method fails if bCapLintener or b-CAP server which is in the machine of the connection destination is not activated.

The parameter required for the provider that starts remotely is specified in the option character string of this method. The list specified in the option character string is shown as follows.

Table2-3 Option character string of CaoWorkspace::AddController

Option	Meaning
Server [=<IP address> [:<port number >]]	Specify IP address and the port number of the b-CAP server. (default value: “127.0.0.1:5007")
COM=<COM Port> [:<Baud Rate> [:<Parity> :<DataBits> :<StopBits> [:Flow]]]	COM network transmission setting. When this option is set, Server, MyIP, and the UDP option are disregarded. <COM Port> : COM port number. ‘1'-COM1,‘2'-COM2,… <BaudRate> : Transmission rate. 4800,9600,19200,38400,57600,115200. <Parity> : Parity. ‘N'-NONE,‘E'-EVEN,‘O'-ODD. <DataBits> : Number of data bits. ‘7'-7bit,‘8'-8bit. <StopBits> : Number of stop bits. ‘1'-1bit,‘2'-2bit. <Flow> : Flow control. (Default: 0 Flow control: None) ‘1' ..Xon/Xoff.. ‘2'-hardware - control. Able to specify it by omitting OR.
MyIP[=<local IP address>]	NIC can be selected by specifying IP address by this option when two or more NICs are used. An appropriate address is automatically selected when omitting input. When IP address which is not allocated in a local machine is specified, the error is returned. This option is disregarded at the RS232C connection.
Provider =<Provider name>	Provider name that starts remotely. (default value: Null character string)
Machine[=<machine name>]	Specify when a remote provider is started with a machine different from the WEB server.

	(default value: Null character string)
Option[=<option character string>]	Specify the option character string required for a remote provider. (default value: Null character string)
Message[=<True/False>]	Status of message acquisition. True: Valid the message acquisition (default). False: Invalid the message acquisition.
Interval =<Polling interval>	Specify the message acquisition interval (ms). (default value: 1000 ms)
UDP[=<True/False>]	Network transmission setting by UDP True: UDP False: TCP (default) The maximum size of the packet becomes 488 bytes at the UDP communication.
ConnTimeout=< time-out time >	Specify the time-out time (ms) of connection invocation.(default: 5000 ms) The value of Timeout option is set up, when this option is omitted and Timeout option is specified. The minimum value is 1000 ms. Less than 1000 value is treated as 1000.
Timeout=< time-out time >	Time-out time when sending and receiving. (default: 500 ms)
TORetry=<.Retry frequency>	Retry frequency when UDP is sent and received. 1-7 (Default: 5) Less than one is regarded as one. More than seven is regarded as seven. The time-out response time of UDP is calculated by the following formula Time-out response time = <Timeout>×<TORetry>
WDT=<Watch dog timer interval>	Watch dog timer interval (ms) of b-CAP server (Default: off). If this option is set and a certain command took a long time, b-CAP server sends a special packet (executing notify packet) to b-CAP client (b-CAP provider) every specified interval for resetting the timeout count. The minimum value is 80 (ms) and this option is valid when the server is “bCapService” or “bCapListener” included in “ORiN2 SDK”.
InvokeTimeout=<time-out time>	Specify the time-out time (ms) of command invocation. (default: 180000 ms) When a time-out occurred, the server exits from wait-loop of command invocation. The value must be above 80 ms, and this option is valid

	when the server is “bCapService” or “bCapListener” included in “ORiN2 SDK”.
AsyncCancel[=<True/False>]	Asynchronous cancel mode. True: Asynchronous cancel mode False : Normal mode (Synchronous cancel mode) If true, “ProviderCancel” command and “ProviderClear” command of CaoContoller::Execute() are executed asynchronously. This option is valid only for TCP communication. [Notes] This option can be set in the registry database. To register this option, specify it in the parameter box of “b-Cap” provider by using “CaoConfig” tool. If this option is not specified when calling AddController() function, the registry value is used.
ProtcolVersion[=<version>]	Specification of b-CAP protocol version.(default:0) Set this value to 1 to use b-CAP/ZIP compression communication.
ZIPMode[=<compression level>]	Specification of b-CAP/ZIP compression level.(default：-1) -1 The default compression(set to 6) 0 no compression 1 - 9 compression levels 1 gives best speed, 9 gives best compression. lower gives more speed. higher gives more compression. Set Protocol Version option value to 1 to apply b-CAP/ZIP compression communication.
ZipThreshold[=<compression size threshold>]	Specification of b-CAP/ZIP compression size threshold.Specification size in KBytes. (default：1 = 1 KBytes) ZIP compression will apply when a size of a b-CAP packet exceeds this value.
Debug[=<True/False>]	Specification of debug mode True: Debug mode False: Normal mode The following variables can be used at debug mode. $LAST_SEND_PACKET$ $LAST_RECEIVE_PACKET$
SSL[=<True/False>]	Version 1.3.0. or higher Secure communication by SSL

	True：SSL False：TCP (Default)
Certificate=<certificate file name>	Available only when “SSL” is set to” True” Specify a certificate file name. (Default: null)
PrivateKey=<private key file name>	Available only when “SSL” is set to” True”. Specify a private key file name. (Default: null)
Password=<password>	Available only when “SSL” is set to” True. Specify a password of private key (Default: null)
CA=<CA certificate file name>	Available only when “SSL” is set to” True”. Specify a certificate file name of CA (certificate authority). (Default: null)

The example when the AddController method is executed is shown as follows.

```
AddController
(
        “RC1”,                  // Controller name  ＝  RC1
        “CaoProv.b-CAP”,        // Fixed
        “”,                     // Execute CAP provider in the CAO engine process.
        “Server=10.8.109.116:5007,Provider=CaoProv.DataStore”
                                // Activate DataStore provider
                                // at IP address : 10.8.109.116, port number : 5007
);
```

The following sample program uses SSL for secure communication.

```
AddController
(
        “RC1”,                                  // Controller name  ＝  RC1
        “CaoProv.b-CAP”,                       // Fixed
        “”,             // Execute CAP provider in the CAO engine process.
        “Server=10.8.109.116:5107,Provider=CaoProv.DataStore,” _
        & “SSL,Certificate=C:¥store¥client.pem,PrivateKey=C:¥store¥client.pem,” _
        & “Password=pass,CA=C:¥store¥rootcert.pem”
                                                // Activate DataStore provider
                                                // at IP address：10.8.109.116, port number：5107
                                                // Certificate file name = C:¥store¥client.pem
                                                // Private key file name = C:¥store¥client.pem
                                                // Password = pass
                                                // CA certificate file name = C:¥store¥rootcert.pem
);
```

[Reference]

When the AddController method is executed, two of the b-CAP’s function, (1) Service_Start and (2) Controller_Connect, are called in this order. Moreover, when the CaoController object made by AddController is deleted, (3) Controller_Disconnect and (4) Two functions of Service_Stop are called in this order.

2.2.2. Method properties other than AddController

As for the b-CAP provider, all the methods and properties of a controller, robots, files, tasks, variables, and extension board classes are mounted. The method and the properties other than AddController(2.2.1) execute the method and the property of the same name with CAO of the server.

2.3. Variable list

There is no variable peculiar to the b-CAP provider.

2.4. Error code

In the b-CAP provider, there is no peculiar error code. About the ORiN2 commonness error, please refer to the chapter of the error code of "ORiN2 Programming guide”.

3. b-CAP listener

3.1. Outline

The b-CAP listener receives the b-CAP message, and executes corresponding CAO method. When communicating by b-CAP, it is necessary to activate bCapListener beforehand on the server side.

There are two types of b-CAP listeners as follows.

Table3-1 b-CAP listener program

Program name	Program type
bCapListener.exe	Console application
bCapService.exe	Windows service program

When the b-CAP listener is executed, CAO should set the access right for the start user of the b-CAP listener.

3.2. bCapListener.exe

3.2.1. Setting of execution parameter

bCapLitener.exe is set in the command line when starting.

The following commands are available in bCapListener.exe. These commands can be arbitrarily combined.

```
bCapListener.exe [/P:Port number] [/C:Number of maximum clients] [/[-]K]
```

There is no distinction between the capital letter and the small letter in the command option. Words in the square brackets "[ ]" are omittable.

Following shows the details of each command.

/P: Port number

Specify the port number of TCP or UDP used to communicate with the server. (Default: 5007)

※Default value at the SSL secure communication is 5107.

/C: Number of maximum clients

Specify the number of clients that can be connected with the server at the same time. (Default: 10)

/R: Process priority

Specify process priority (0: IDLE, 1: NORMAL, 2: HIGH, 3: REALTIME) . (Default: 2)

/T: Time-out time

Specify the communication time-out time. (Default: 500)

/U

Communicate with UDP. (Default: TCP)

/K

Use the keep alive function of TCP. (Default: ON)

This function is set to invalid when specified it with hyphen (E.g.: “/-K").

/S

Perform SSL communication. (Default: TCP)

/CF:Certificate file name

Specify a certificate file name. (Default: null)

※Only for SSL communication

/PK:Private key file name

Specify a private key file name. (Default: null)

※Only for SSL communication

/PW: Password

Specify a password for private key. (Default: null)

※Only for SSL communication

/CA: CA(certificate authority) certificate file name

Specify a certificate file name of CA. (Default: null)

※Only for SSL communication

/CRL:CRL (Certification Revocation List) file name

Specify CRL file name. (Default: null)

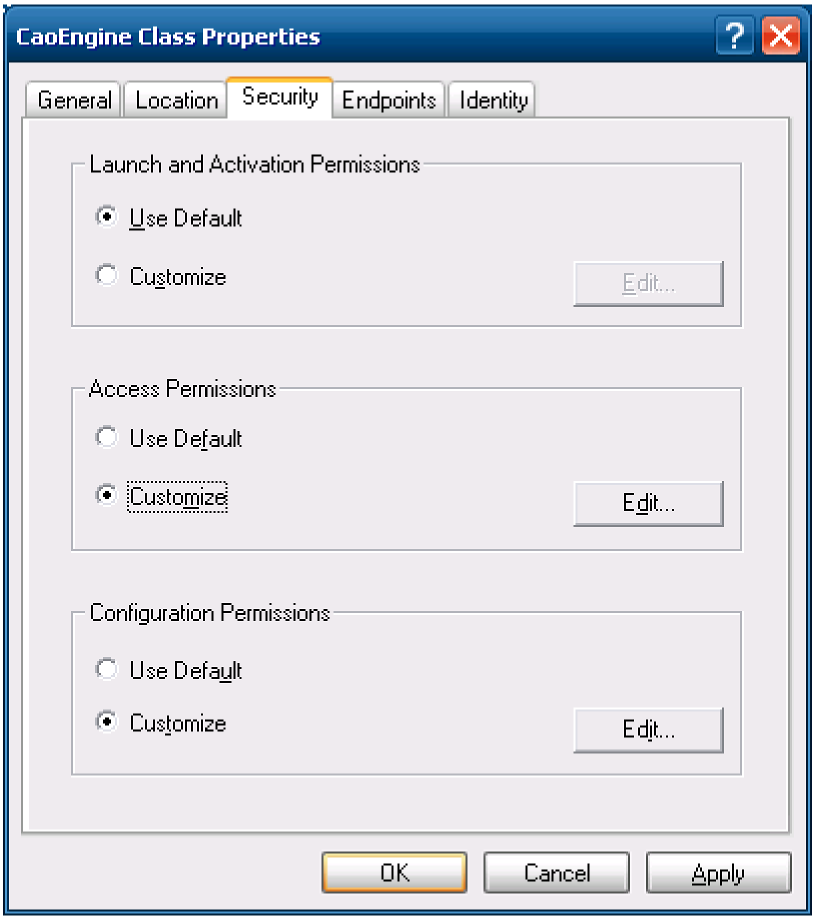
※Only for SSL communication

3.3. bCapService.exe

3.3.1. Security setting

bCapService is started by the system user.. Therefore, system user's access right needs to be set to CAO. Set-up process is as follows.

1. Execute "dcomcnfg" by the command prompt.
2. Select the following destination from the tree of "Component service" window.
   Console route → Component service → Computer → Microcomputer pewter →DCOM composition
3. Select CAO from the list then select "Operation (A)"from the Menu →Select "Property (R)".
4. Select security tab on the property screen of CAO then check "Customize (M)"of the access permission, and click "Edit (D)".



Launch and Activation Permissions

● Use Default

○ Customize Edit...

Access Permissions

○ Use Default

● Customize Edit...

Configuration Permissions

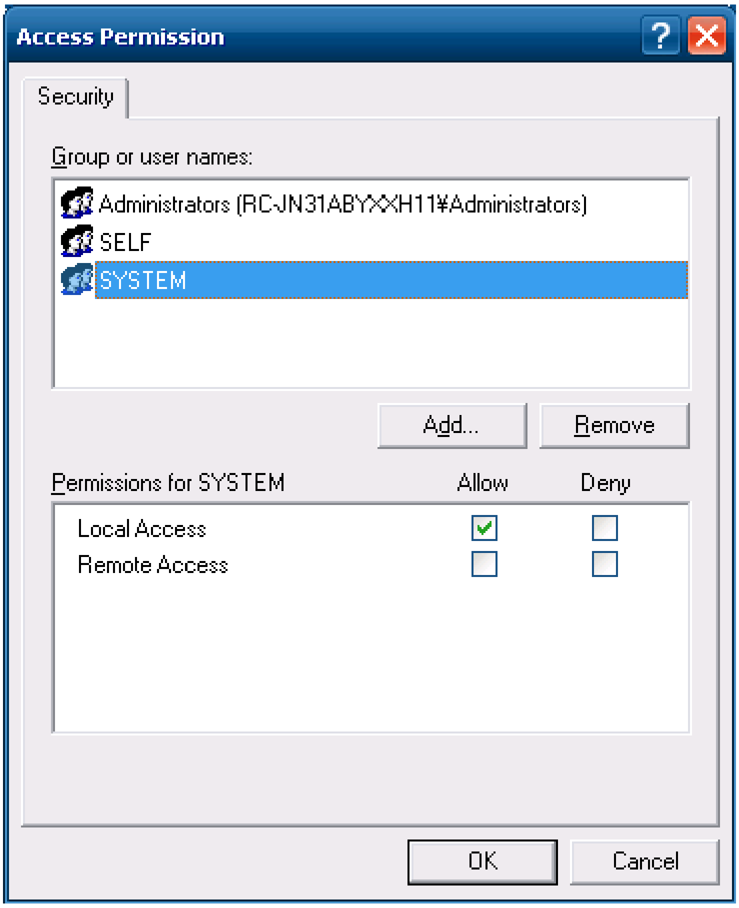
○ Use Default

● Customize Edit...

Figure3-1 DCOM setting screen of CAO (security tab)

5. Click "Add (D)" button to add "SYSTEM" user of a local machine[4].
6. Check the "Local Access" box of the SYSTEM user then click OK button.

[4] This process is not necessary when SYSTEM user exists



Access Permission

Security

Group or user names:

Administrators (RC-JN31ABYXXH11¥Administrators)

SELF

SYSTEM

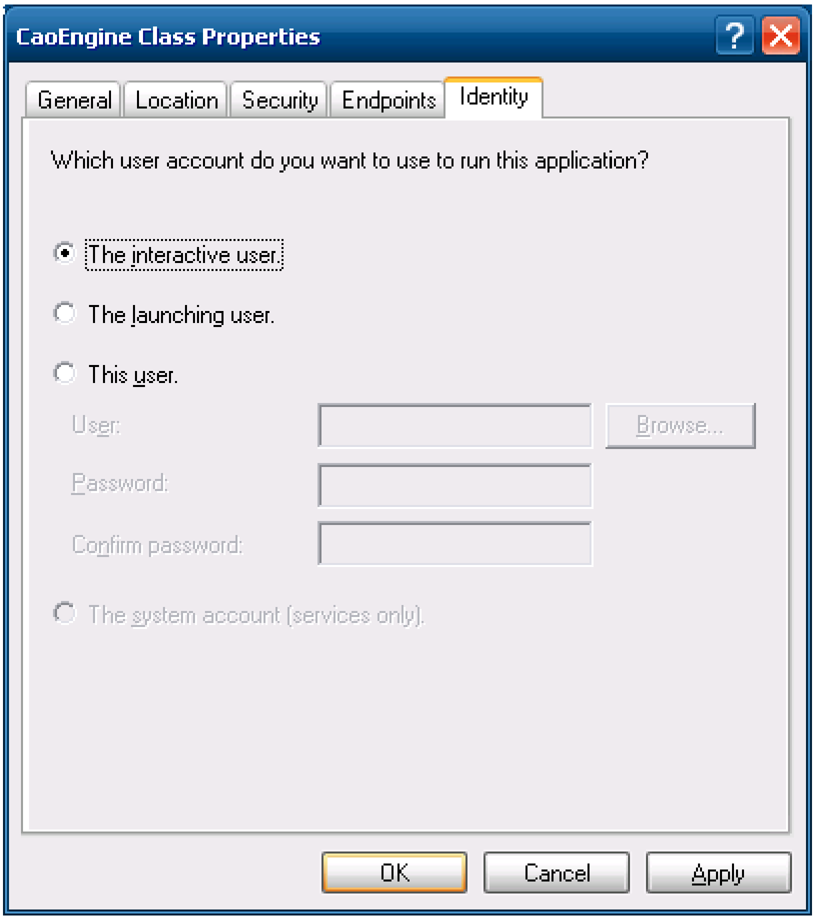
Permissions for SYSTEM

Permissions for SYSTEM	Allow	Deny
Local Access	☑	□
Remote Access	□	□

Figure3-2 Setting screen of access permission of CAO

7. Select ID tab on the property screen of CAO, and check " The interactive user (I)"[5].

[5] Please note that if "The interactive user" is checked, two CAO processes (log-in user and system user) are activated when another client starts up CAO. That causes the incorrect communication and doubles the load.



Which user account do you want to use to run this application?

● The interactive user.

○ The launching user.

○ This user.

User:

Password:

Confirm password:

○ The system account (services only).

Figure3-3 DCOM setting screen of CAO (ID tab)

8. Click OK button of the property screen of CAO.

3.3.2. Setting of execution parameter

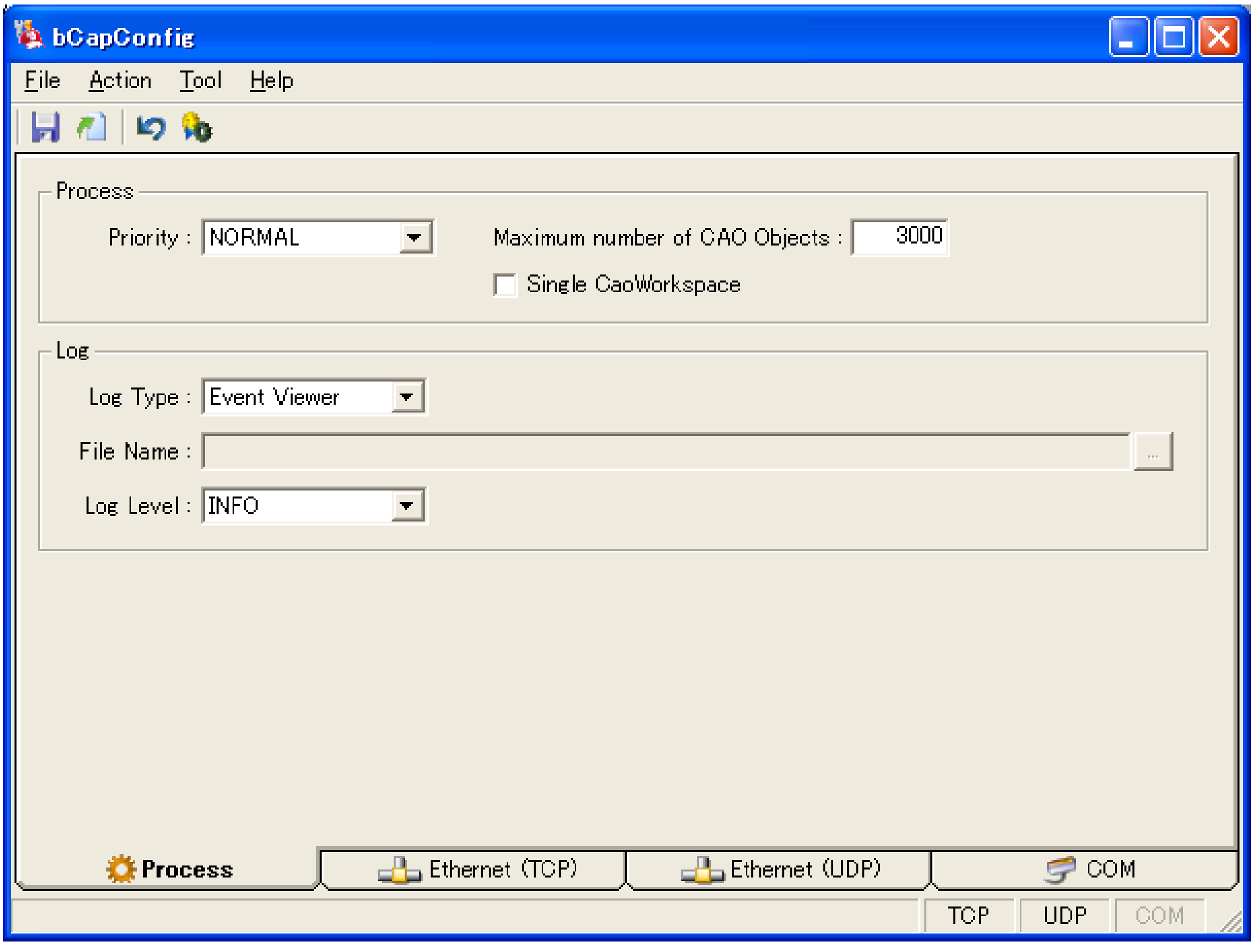
For the setting of bCapService.exe, bCapConfig.exe is used. The setting information is registered in the registry.

Please refer to 4 for details of bCapConfig.exe.

4. bCapConfig

4.1. Outline

b-CAP Configuration Manager (execution file name: bCapConfig.exe. hereafter, bCapConfig) is a tool that configures the network transmission setting of bCapService in the machine. Information configured in this step is recorded in the registry and read when bCapService starts.



Priority : NORMAL

Maximum number of CAO Objects : 3000

☐ Single CaoWorkspace

Log Type : Event Viewer

File Name :

Log Level : INFO

Figure4-1 bCapConfig screen

4.2. Manner of operation

4.2.1. Menu

4.2.1.1. File menu

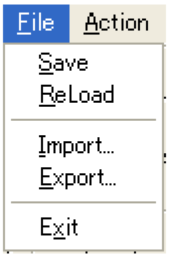


Figure4-2 File menu

[Save] – Save

Set the information of bCapConfig to the registry.

[Reload] – Reload

Acquire the information of bCapConfig from the registry.

[XML import] – Import

Acquire the information of bCapConfig from the XML file.

[XML export] – Export

Set the information of bCapConfig to the XML file.

[Exit] – Exit

Exit bCapConfig.

4.2.1.2. Action menu

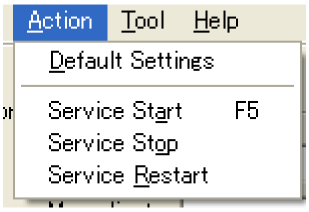


Figure4-3 Action menu

[Default setting] – Default Setting

Change the current display back to the default.

[Service start] – Service Start

Start CAO service. This menu is available only when CAO is registered as a service.

[Service stop] – Service Stop

Stop CAO service. This menu is available only when CAO is registered as a service

[Service restart] – Service Start

Restart CAO service. This menu is available only when CAO is registered as a service.

4.2.1.3. Tool menu

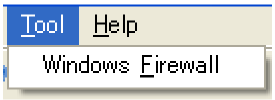


Figure4-4　Tool menu

[Windows firewall setting] – Windows Firewall

Display the setting screen of the Windows firewall.

4.2.1.4. Help menu

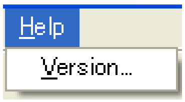


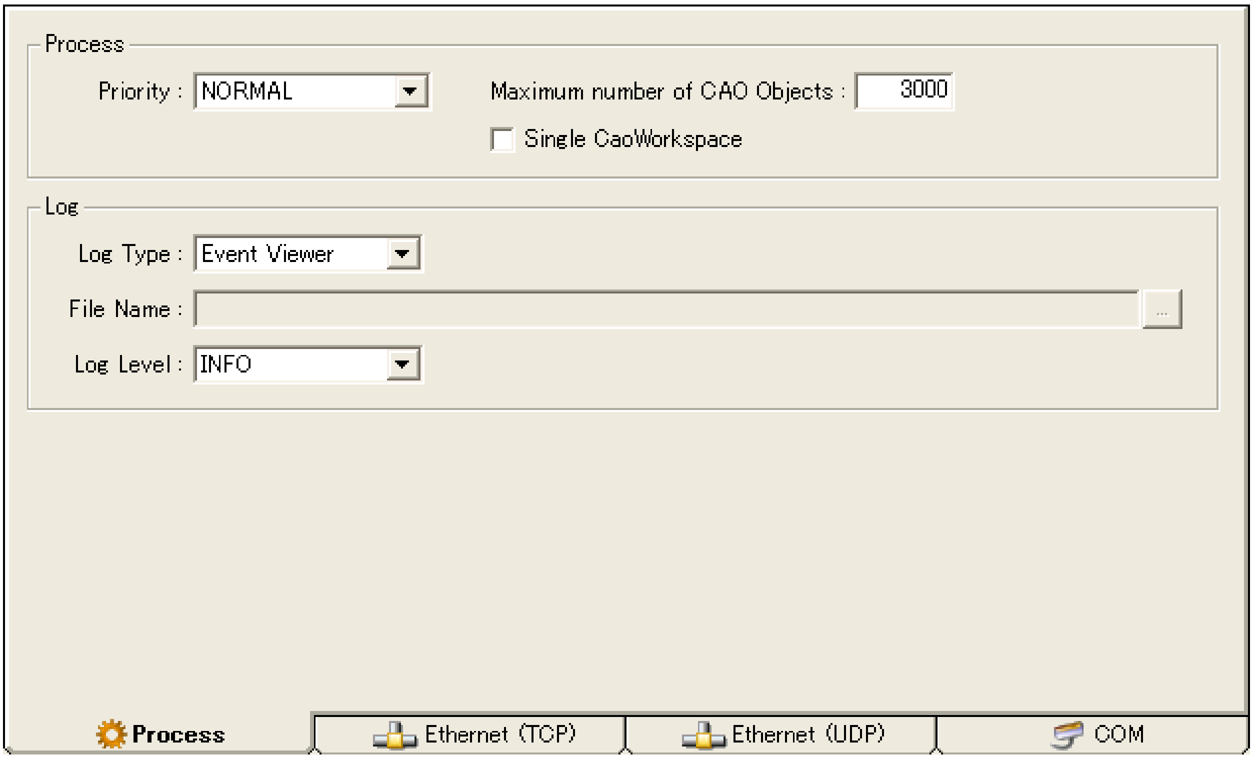
Figure4-5　Help menu

[Version] –**Version**

Display the version information of bCapConfig.

4.2.2. Tab input

4.2.2.1. Process tab



Process

Priority : NORMAL

Maximum number of CAO Objects : 3000

□ Single CaoWorkspace

Log

Log Type : Event Viewer

File Name :

Log Level : INFO

Figure4-6 Process tab

[Process priority] – Process Priority

Set the process priority of the CAO engine. The priority-order is as follows.

REAL TIME > HIGH > **NORMAL** > IDEL

[Single workspace use] – Single CaoWorkspace

Select whether to create the workspace object when bCapService creates the controller object.

Individual workspace of each controller is created when there is no check in this box.

All controllers are created in the same workspace when the box is checked

[Log type] - Log Type

Select the output type of the log of CAO.exe. Following output types of the log are available.

The table 4-1 describes the configurable items.

Table4-1 Log type

At the output destination	Remarks
Console	Output to the console.
Message Box	Output to the message box (Service starts).
Event Viewer	Output to the event viewer (Service starts).
Debug Viewer	Debug output.
Text File	Output to the specified text file.

[Log output file name] – File Name

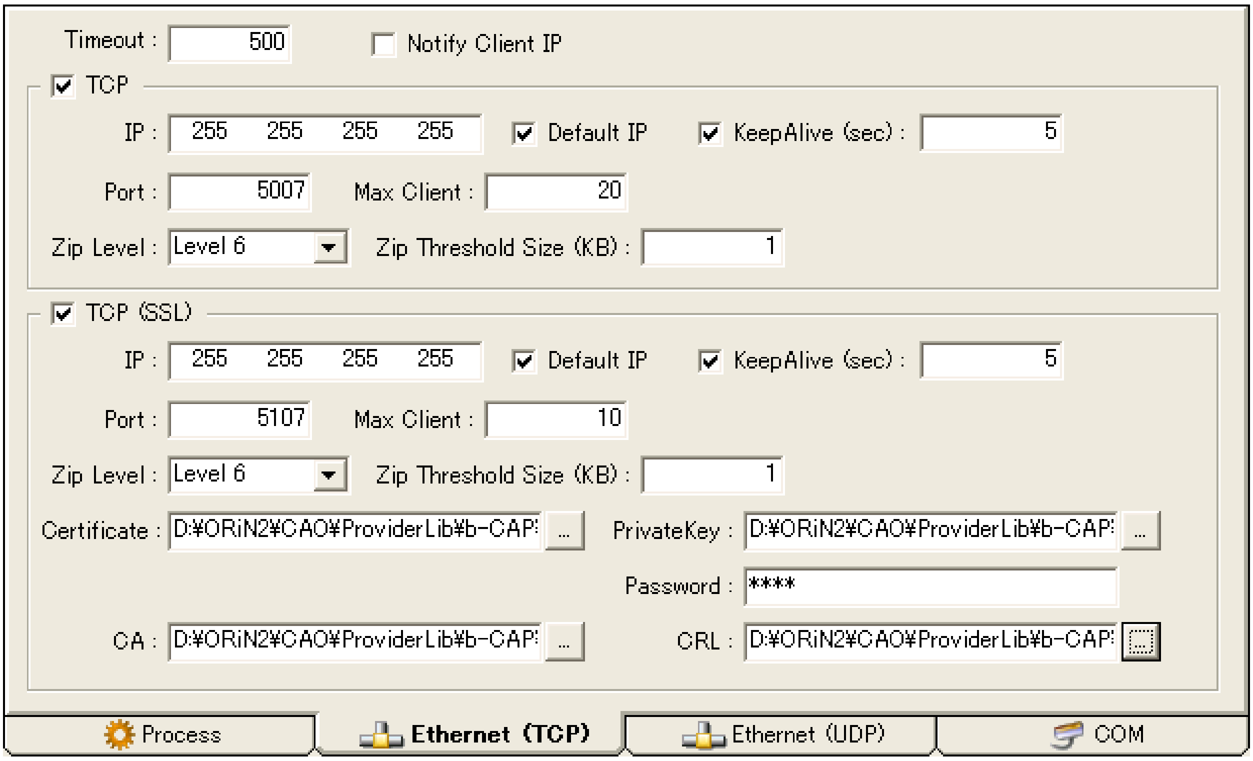
When the log type is a text file, the file path is specified here.

[Log level] – Log Level

The output level of the log is set (Any log of higher level than the setting is output). The log level can be chosen from five levels shown below. The severity is the highest in "FATAL" and the lowest in"DEBUG" in order. Default setting is "INFO".

FATAL > ERROR > WARN > **INFO** > DEBUG

4.2.2.2. Ethernet (TCP) tab



Timeout : 500 □ Notify Client IP

☑ TCP

IP : 255 255 255 255 ☑ Default IP ☑ KeepAlive (sec) : 5

Port : 5007 Max Client : 20

Zip Level : Level 6 Zip Threshold Size (KB) : 1

☑ TCP (SSL)

IP : 255 255 255 255 ☑ Default IP ☑ KeepAlive (sec) : 5

Port : 5107 Max Client : 10

Zip Level : Level 6 Zip Threshold Size (KB) : 1

Certificate : D:¥ORiN2¥CAO¥ProviderLib¥b-CAP! ... PrivateKey : D:¥ORiN2¥CAO¥ProviderLib¥b-CAP! ...

Password : ****

CA : D:¥ORiN2¥CAO¥ProviderLib¥b-CAP! ... CRL : D:¥ORiN2¥CAO¥ProviderLib¥b-CAP! ...

Figure4-7 Ethernet (TCP) tab

[TCP setting] – TCP

Specify the b-CAP/TCP network transmission setting. bCapService.exe. does not execute b-CAP/TCP communication if this "TCP" check box is empty.

[Setting of IP address for TCP] – TCP → IP

Specify IP address used when b-CAP/TCP communication. When DefaultIP is checked, this setting is disregarded.

[Setting of Default IP for TCP] – TCP → DefaultIP

Specify whether to use default IP address when b-CAP/TCP communication. When this setting is checked, the setting of IP address is disregarded.

[Setting of port number for TCP] – TCP → Port

Specify the TCP port number used when b-CAP/TCP communication.

[Number of TCP maximum clients] – TCP → Max client

Specify the number of clients that can be connected with the server at the same time when b-CAP/TCP communication.

[Keep alive setting for TCP] – KeepAlive

Specify the KeepAlive time of b-CAP/TCP communication. If this option was disabled, KeepAlive function is not activated.

[Compression level setting for TCP] – TCP → Zip Level

Specify a compression level for b-CAP/TCP communication.

[Compression threshold size setting for TCP] – TCP → Zip Threshold Size

Specify a compression threshold size for b-CAP/TCP communication.

[TCP (SSL) setting] – TCP (SSL)

Specify b-CAP/TCP (SSL) communication setting. bCapService.exe does not perform b-CAL/TCP (SSL) communication if this item is not checked.

[IP address setting for TCP (SSL)] – TCP (SSL) → IP

Specify IP address for b-CAP/TCP (SSL) communication. This setting will be ignored when DefaultIP is checked.

[Default IP setting for TCP (SSL)] – TCP (SSL) → DefaultIP

Specify whether to use default IP address at b-CAP/TCP (SSL) communication. The setting of IP address will be ignored when this item is checked.

[Setting of port number for TCP (SSL)] – TCP (SSL) → Port

Specify TCP port number that is used at b-CAP/TCP (SSL) communication.

[Number of TCP (SSL) maximum clients] – TCP (SSL) → Max client

Specify the number of clients that can be connected with the server at the same time when b-CAP/TCP (SSL) communication.

[Keep alive setting for TCP (SSL)] – TCP (SSL) → KeepAlive

Specify the KeepAlive time of b-CAP/TCP (SSL) communication.

KeepAlive is not used when this item is not checked.

[Compression level setting for TCP (SSL)] – TCP (SSL) → Zip Level

Specify a compression level for b-CAP/TCP (SSL) communication.

[Compression threshold size setting for TCP (SSL)] – TCP (SSL) → Zip Threshold Size

Specify a compression threshold size for b-CAP/TCP (SSL) communication.

[Certificate file name setting] – TCP (SSL) → Certificate

Specify a certificate file name used at b-CAP/TCP (SSL) communication

[Private key file name setting] – TCP (SSL) → PrivateKey

Specify a private key file name used at b-CAP/TCP (SSL) communication.

[Private key password setting] – TCP (SSL) → Password

Specify a password for private key used at b-CAP/TCP (SSL) communication.

[CA file name setting] – TCP (SSL) → CA

Specify a CA (Certificate Authority) certificate file name used at b-CAP/TCP (SSL) communication.

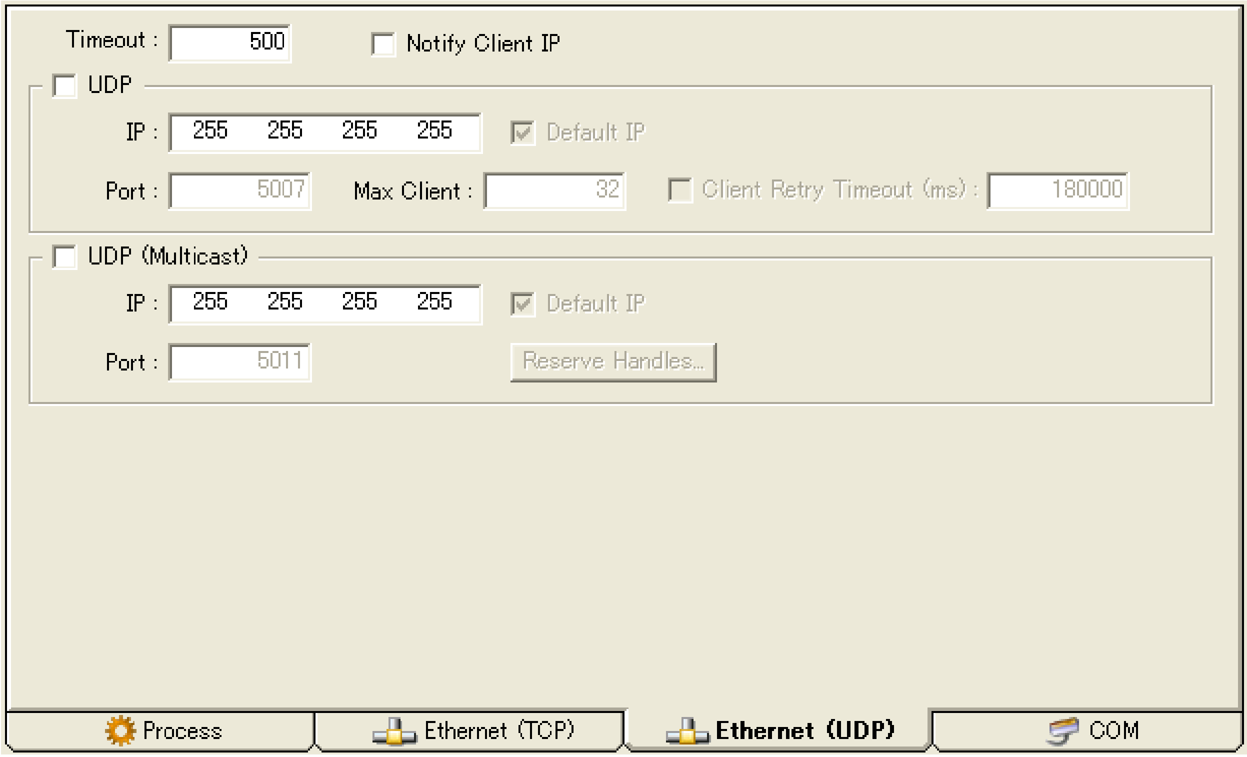
[CRL file name setting] – TCP (SSL) → CRL

Specify a CRL (Certificate Revocation List) file name used at b-CAP/TCP (SSL) communication.

[Setting of communication time-out time] – Timeout

Specify the time-out time used when communication.

4.2.2.3. Ethernet (UDP) tab



Timeout : 500

□ Notify Client IP

□ UDP

IP : 255 255 255 255

☑ Default IP

Port : 5007

Max Client : 32

□ Client Retry Timeout (ms) : 180000

□ UDP (Multicast)

IP : 255 255 255 255

☑ Default IP

Port : 5011

Figure4-8 Ethernet (UDP) tab

[UDP setting] – UDP

Specify the b-CAP/UDP network transmission setting. This setting communicates and when the check is not done, b-CAP/UDP doesn't communicate bCapService.exe. bCapService.exe. does not execute b-CAP/UDP communication if the box is checked

[Setting of IP address for UDP] – UDP → IP

Specify IP address used when b-CAP/UDP communication. When DefaultIP is checked, this setting is disregarded.

[Setting of default IP for UDP] – UDP → DefaultIP

Specify whether to use the default IP address when b-CAP/UDP communication. When this setting is checked, the setting of IP address is disregarded.

[Setting of port number for UDP] – UDP → Port

Specify the UDP port number used when b-CAP/UDP communication.

[Setting of maximum UDP connection] – UDP → Max Client

Specify the maximum number of client that can be connectable simultaneously at the b-CAP/UDP connection.

[Maximum holding time of UDP retry count result] – UDP → Client Retry Timeout (ms)

Specify the maximum holding time of the execution result that is returned when b-CAP/UDP communication is retried.

If retry is occurred within this holding time, a command will not be executed and then an execution result of just before will be returned.

If the duration of the communication timeout of client application is shorter than this setting, command can be executed by a retry at unexpected timing.

[UDP (Multicast) setting] – UDP (Multicast)

Specify b-CAP/UDP (Multicast) communication setting. bCapService.exe does not perform b-CAP/UDP (Multicast) communication if this item is not checked.

[IP address setting for UDP (Multicast)] – UDP (Multicast) → IP

Specify IP address for b-CAP/UDP (Multicast) communication. This setting will be ignored when DefaultIP is checked.

[Default IP setting for UDP (Multicast)] – UDP (Multicast) → DefaultIP

Specify whether to use default IP address for b-CAP/UDP (Multicast) communication. The setting of IP address will be ignored when this item is checked.

[Setting of port number for UDP (Multicast)] – UDP (Multicast) → Port

Specify UDP port number that is used at b-CAP/UDP (Multicast) communication.

[Reserved object handle setting for UDP (Multicast)] – UDP (Multicast) → Reserve Handles…

Specify a reserved object handle for b-CAP/UDP (Multicast) communication.

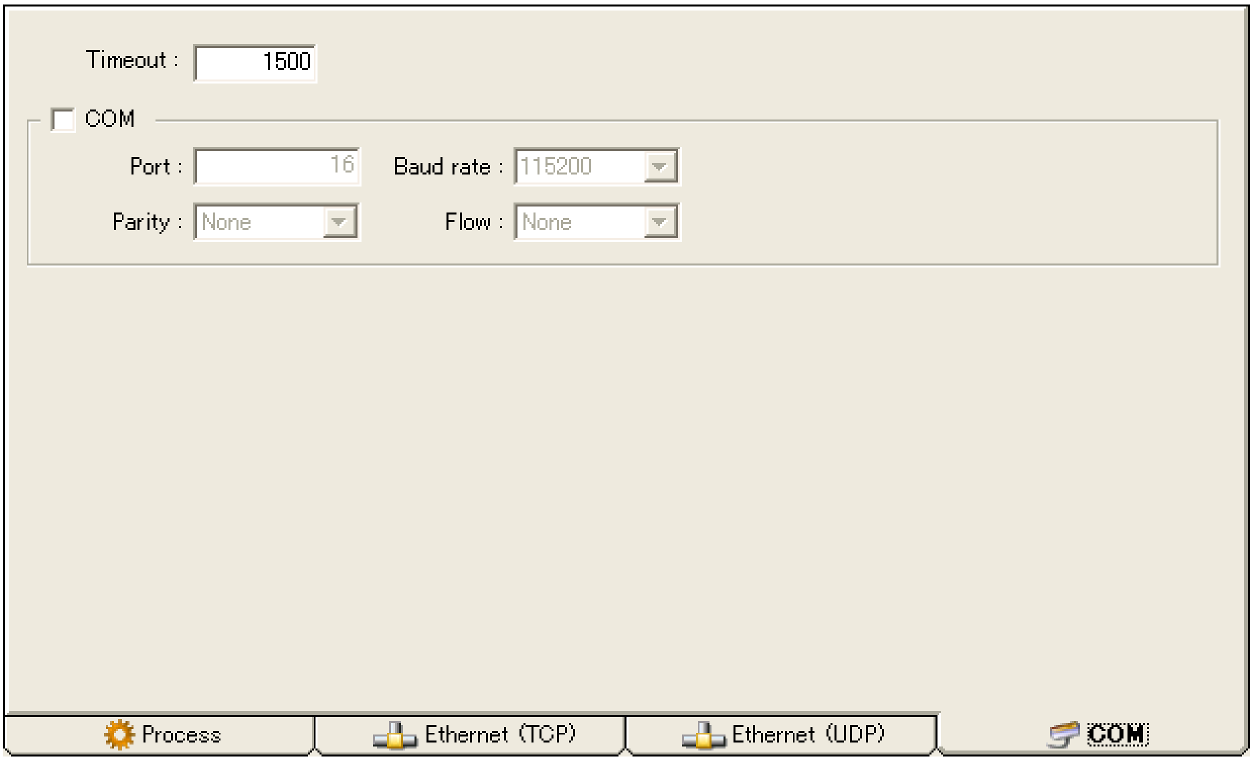
When you click this button, the reserved object handle input dialog will be displayed.

For details, refer to 4.2.2.5.

[Setting of communication time-out time] – Timeout

Specify the time-out time used when communication.

4.2.2.4. COM tab



Timeout : 1500

□ COM

Port : 16 Baud rate : 115200

Parity : None Flow : None

Figure4-9 COM tab

[COM setting] – COM

Specify the b-CAP/COM network transmission setting. bCapService.exe. does not execute b-CAP/COM communication when the checkbox is empty.

[Setting of port number for COM] – COM → Port

Specify the COM port number used when b-CAP/COM communication.

[Setting of baud rate for COM] – COM → Baud rate

Specify the baud rate used when b-CAP/COM communication.

[Setting of parity for COM] – COM → Parity

Specify the type of the parity used when b-CAP/COM communication.

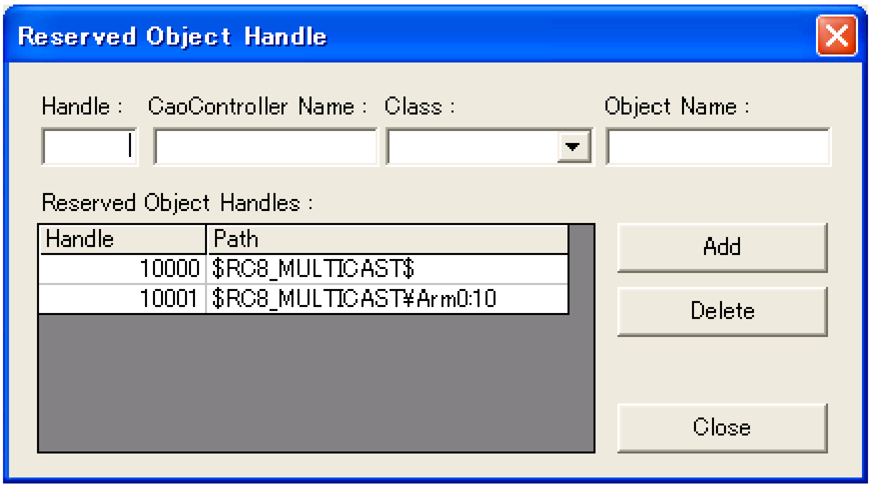
[Setting of flow control for COM] – COM → Flow

Specify the type of the flow control used when b-CAP/COM communication.

[Setting of communication time-out time] – Timeout

Specify the time-out time used when communication.

4.2.2.5. Reserved Object Handle input dialog



Handle	Path
10000	$RC8_MULTICAST$
10001	$RC8_MULTICAST¥Arm0:10

Figure 4-10 Reserved object handle input dialog

[Handle number] – Handle

Specify a handle number to add.

[CAO controller name] – CaoController Name

Specify CAO controller name to add.

[Class] – Class

Select a class to add from the table 4-2.

Table 4-2 Class

Value	Class
6	Extension
8	File
10	Robot
12	Task
14	Variable
16	Command

[Object name] – Object

Specify an object name to add.

[Reserved object handle list] – Reserved Object Handles

Display the reserved object handle list.

[Add] – Add

Add an object handle being entered to the reserved object handle list.

[Delete] – Delete

Delete a currently selected reserved object handle from the reserved object handle list.

[Close] – Close

Close this dialog.

5. Sample program

The sample program below shows the way of activating the DataStore provider with the server (IP address: 10.8.109.116) and setting / acquiring the value to/from the variable.

At this time, it is assumed that bCapListerner has started with the server.

List 5-1 **Sample.frm**

```
Private eng As CaoEngine
Private ctrl As CaoController
Private var As CaoVariable

Private Sub Form_Load()

  Dim ws As CaoWorkspace

  Set eng = New CaoEngine
  Set ws = eng.Workspaces(0)

  ‘Connect to the server.
  Set ctrl = ws.AddController("RC1", _
                              "CaoProv.b-CAP", _
                              "", _
                              "Provider=CaoProv.DataStore,Server=10.8.109.116")

  ‘Acquire the variable
  Set var = ctrl.AddVariable("Var1")

End Sub

'Set the variable
Private Sub Command1_Click()
  var = Text1.Text
End Sub

'Acquisition of variable
Private Sub Command2_Click()
  Text1.Text = var
End Sub
```

The following sample program uses SSL for secure communication.

List 5-2 **Sample2.frm**

```
Private eng As CaoEngine
Private ctrl As CaoController
Private var As CaoVariable

Private Sub Form_Load()

    Dim ws As CaoWorkspace

    Set eng = New CaoEngine
    Set ws = eng.Workspaces(0)

    'Connect to the server
    Set ctrl = ws.AddController("RC1", _
                                "CaoProv.b-CAP", _
                                "", _
                                "Provider=CaoProv.DataStore,Server=10.8.109.116,5107,SSL," _
                                 & “Certificate=C:¥store¥client.pem,” _
                                 & “PrivateKey=C:¥store¥client.pem,” _
                                 & “Password=pass,” _
                                 & “CA=C:¥store¥rootcert.pem”)

    'Acquire the variable
    Set var = ctrl.AddVariable("Var1")

End Sub

'Set the variable
Private Sub Command1_Click()
    var = Text1.Text
End Sub

'Acquisition of variable
Private Sub Command2_Click()
    Text1.Text = var
End Sub
```

6. Appendix

6.1. Secure communication via SSL

The following shows how to set up secure communication with SSL.

Here, we will explain the following:

Overview

Creating Required Files

Creating Root CA Certificates and Certificate Revocation Lists

Creating a Server Certificate and Private Key

Creating a Client Certificate and Private Key

6.1.1. Overview

This section describes how to create a root CA certificate and a certificate revocation list, and to create a server certificate and a client certificate signed by that certificate.

Use openssl.exe for creation.

The following assumes that ORiN2 is installed in C:¥.

Modify the parameters accordingly.

For more information about the commands, see the help for openssl.exe.

To execute the command in this chapter, set the directory of C:¥ ORiN2¥Tools¥OpenSSL¥Bin¥ openssl.cnf (line 45).

```
dir                 = . /demoCA                # Where everything is kept
→
dir                 = .                # Where everything is kept
```

And the explanation is made with the corrections.

When executing the command, use any folder that outputs the necessary files.

The explanation in this chapter is intended for use in validation.

6.1.2. Creating Required Files

This section describes the procedure for creating files that are required for input and output when the command is executed.

6.1.2.1. Creating a.txt index

Create a database index file for the signed certificate.

Run the following command:

```
type nul > index.txt
```

6.1.2.2. Creating a crlnumber

Create the next CRL number file.

Run the following command:

```
echo 00 > crlnumber
```

6.1.2.3. Creating a serial

Create a Certificate Serial Number File.

Run the following command:

```
echo 01 > serial
```

6.1.3. Creating Root CA Certificates and Certificate Revocation Lists

Create a root CA certificate and a certificate revocation list to sign the server certificate and client certificate.

rootcert.pem => Root CA certificate

rootcrl.crl => Certificate revocation list

It becomes.

6.1.3.1. Certificate Signing Request and Private Key Generation for the Root CA

Create the certificate signing request and private key required to create the root CA certificate.

rootreq.csr => Root CA Certificate Signing Requests

rootkey.pem => Root CA private key

It becomes.

Example)

- Expiration date: 10 years (3650 days)
- Output certificate signing request file: rootreq.csr
- Issuer: Country-JP, City-Your Loccality, State-Your State, Organization-Your Organization, Department-Your Division, Name-Your CA
- Cryptographic key strength: RSA (1024 bits)
- Hash function :SHA1
- Output private key file: rootkey.pem
- Password: password

Run the following command:

```
C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.exe req -config
"C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.cnf" -new -days 3650 -out rootreq.csr -subj "/C=JP/L=Your
Locality/ST=Your State/O=Your Organization/OU=Your Division/CN=Your CA" -newkey rsa:1024
-sha1 -keyout rootkey.pem -passout pass:password
```

6.1.3.2. Creating a Root CA Certificate

Create a root CA certificate using the certificate signing request and private key that you created.

rootcert.pem => Root CA certificate

It becomes.

Example)

- Input Certificate Signing Request File: rootreq.csr
- Hash function :SHA1
- Input private key file: rootkey.pem

- Output root CA file: rootcert.pem
- Password: password
- Expiration date: 10 years (3650 days)

Run the following command:

```
C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.exe x509 -req -in rootreq.csr -sha1 -extfile
"C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.cnf" -extensions v3_ca -signkey rootkey.pem -CAserial serial
-out rootcert.pem -passin pass:password -days 3650
```

6.1.3.3. Generation of CRLs

Create a certificate revocation list using the root CA certificate and the root CA private key that you created.

rootcrl.crl => Certificate revocation list

It becomes.

Example)

- Expiration date: 1 year (365 days)
- Certificate Revocation List File:rootcrl.crl
- Input root CA file:rootcert.pem
- Input root CA private key file: rootkey.pem
- Password: password

Run the following command:

```
C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.exe ca -config
"C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.cnf" -gencrl -crldays 365 -out rootcrl.crl -cert rootcert.pem
-keyfile rootkey.pem -passin pass:password
```

6.1.4. Creating a Server Certificate and a Server Private Key

Create a server certificate and a server private key using the root CA certificate and the root CA private key.

servercert.pem => ServerCertificate

serverkey.pem => Private key

It becomes.

6.1.4.1. To create a server certificate signing request and private key

Create the certificate signing request and private key required to create the server certificate.

serverreq.csr => Server Certificate Signing Requests

serverkey.pem => Private key

It becomes.

Example)

- Output certificate signing request file: serverreq.csr
- Issuer: Country-JP, City-Your Loccality, State-Your State, Organization-Your Organization, Department-Your Division, Name-bCAP
- Cryptographic key strength: RSA (1024 bits)
- Hash function :SHA1
- Output private key file: serverkey.pem

- Password: password

Run the following command:

The name of the issuer of the server certificate must be "bCAP".

```
C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.exe req -config
"C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.cnf" -new -out serverreq.csr -subj "/C=JP/L=Your
Locality/ST=Your State/O=Your Organization/OU=Your Division/CN=bCAP" -newkey rsa:1024 -sha1
-keyout serverkey.pem -passout pass:password
```

6.1.4.2. Creating a server certificate

Create a server certificate using the certificate signing request and private key that you created.

servercert.pem => ServerCertificate

It becomes.

Example)

- Input Certificate Signing Request File: serverreq.csr
- Output server certificate file: servercert.pem
- Password: password
- Expiration date: 10 years (3650 days)
- Certificate revocation list expiration date: 10 years (3650 days)
- Root CA certificate file: rootcert.pem
- Root CA private key file: rootkey.pem
- Output directory:. (dot)

Run the following command:

```
C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.exe ca -config
"C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.cnf" -extensions usr_cert -batch -in serverreq.csr -out
servercert.pem -passin pass:password -days 3650 -crldays 3650 -cert rootcert.pem -keyfile rootkey.pem
-outdir .
```

6.1.5. Creating a Client Certificate and a Client Private Key

Create a client certificate and client private key using the root CA certificate and the root CA private key.

clientcert.pem => Client certificate

clientkey.pem => client-private key

It becomes.

6.1.5.1. Client Certificate Signing Request and Private Key Generation

Create the certificate signing request and private key required to create the client certificate.

clientreq.csr => Client Certificate Signing Requests

clientkey.pem => client-private key

It becomes.

Example)

- Output certificate signing request file: clientreq.csr
- Issuer: Country-JP, City-Your Loccality, State-Your State, Organization-Your Organization,

Department-Your Division, Name-Your CommonName

- Cryptographic key strength: RSA (1024 bits)
- Hash function :SHA1
- Output private key file: clientkey.pem
- Password: password

Run the following command:

```
C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.exe req -config
"C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.cnf" -new -out clientreq.csr -subj "/C=JP/L=Your
Locality/ST=Your State/O=Your Organization/OU=Your Division/CN=Your CommonName" -newkey
rsa:1024 -sha1 -keyout clientkey.pem -passout pass:password
```

6.1.5.2. Creation of client certificates

Create a client certificate using the certificate signing request and private key that you created.

clientcert.pem => ServerCertificate

It becomes.

Example)

- Input Certificate Signing Request File: clientreq.csr
- Output server certificate file: clientcert.pem
- Password: password
- Expiration date: 10 years (3650 days)
- Certificate revocation list expiration date: 10 years (3650 days)
- Root CA certificate file: rootcert.pem
- Root CA private key file: rootkey.pem
- Output directory:. (dot)

Run the following command:

```
C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.exe ca -config
"C:¥ORiN2¥Tools¥OpenSSL¥Bin¥openssl.cnf" -extensions usr_cert -batch -in clientreq.csr -out
clientcert.pem -passin pass:pass -days 3650 -crldays 3650 -cert rootcert.pem -keyfile rootkey.pem
-outdir .
```

6.1.6. Settings and connections for each application

In the creation

- Root CA certificate file (rootcert.pem)
- Server/client certificate file (servercert.pem/clientcert.pem)
- Server/client private key file and its password (serverkey.pem/clientkey.pem) (the password you entered when you requested the server/client to sign the certificate and create the private key)
- Certificate Revocation List File (Servers Only) (rootcrl.crl)

Specify to launch and connect to each app.

For the setting method

- bCapListener.exe 3.2.1

- bCapService.exe => bCapConfig 4.2.2.2
- CaoProvBCAP 2.2.1

Refer to the setting method.