POP3 provider

Version 1.0.0

User’s guide

January 15, 2015

[Remarks]

[Revision history]

Version	Date	Content
1.0.0	2015-01-15	First edition

Contents

1. Introduction

POP3 provider is a CAO provider that enables to receive e-mails based on the POP3 protocol processing. With this provider, users can download e-mails from UIDL command-supported POP3 server to a client computer and also can delete such e-mails.

This document describes the outline of POP3 provider and the implemented CAO interface (function specification).

2. Outline of provider

2.1. Installation

POP3 provider module consists of the following DLL. You do not have to install it if POP3 provider is installed by the installer of ORiN2 SDK. To install it manually, use the information of Table 2-1.

Table 2-1 POP3 provider

File name	CaoProvPOP3.dll
ProgID	CaoProv.POP3
Registration	regsvr32 CaoProvPOP3.dll
Deregistration	regsvr32 /u CaoProvPOP3.dll

POP3 provider uses POCO C++ Libraries to connect to POP3 server for mail reception

2.2. Outline

By connecting to UIDL command-supported POP3 server, POP3 provider receives the mail list and mail messages and deletes mails.

2.3. Method and properties

2.3.1. CaoWorkspace::AddController

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

For bstrOption, enter a domain name (or an IP address) of a server, port number, user name, password, and timeout period.

Specify a content with the form of [item]=[content] with a comma delimiter.

Example: HOSTNAME=127.0.0.1,USER=Test,PASS=Test,PORT=110,TIMEOUT=10000

< bstrCtrlName >	:	[in] Controller name
< bstrProvName >	:	[in] Provider name. Fixed to ”CaoProv.POP3”
<bstrPcName>	:	[in] Computer name where provider runs (not used)
<bstrOption>	:	[in] Option character strings

HOSTNAME	:	Specify a connection destination. (required) A domain name or IP address of POP3 server to connect. Example) HOSTNAME=127.0.0.1
USER	:	Specify a user name. (required)
PASSWORD	:	Specify a password. (required)
PORT	:	Specify a port number.(optional) 110 is entered if it is omitted.
TIMEOUT	:	Specify a timeout period by milliseconds.(optional) Enter a value within the range of integer type. 0 (infinite wait) is entered if it is omitted.

2.3.2. CaoController::AddFile method

Syntax AddFile (<bstrName:BSTR>, <bstrOption:BSTR>)

Analyze a mail corresponding to the specified UDIL and then create a CaoFile to store mail information. For bstrOption, specify a UDIL of a mail..

Specify a content with the form of [item]=[content].

Example: UIDL=TestUIDL

< bstrName >	:	[in] File name

<bstrOption> : [in] Option character strings

UIDL : Specify a UIDL of a mail. (required)

2.3.3. CaoFile::get_Value property

Obtain contents of a mail that corresponds to a UDIL specified by CaoController::AddFile. Contents of a mail is obtained as arrays of character strings.

Basically, contents of a mail to be obtained is a body part of a mail which Content-Disposition header is other than “attachment” and Content-Type is “text/plain”.

As an exceptional case, if a MIME type is multipart/digest, the body of the part which Content-Disposition header is other than “attachment” and which Content-Type is “message/rfc822” is obtained.

3. Command reference

3.1. Controller class

Table 3-1 CaoController::Execute command list

Command	Description	Page
MessageCount	Obtain the number of emails.	P.8
ListUIDL	Obtain the list of UIDL of emails.	P.8
Size	Obtain the size of an email	P.9
DeleteMessage	Delete an email	P.9
SetTimeout	Set a timeout period	P.9
GetTimeout	Obtain a timeout period	P.10

3.1.1. CaoController::Execute(“MessageCount”) command

Obtain the number of mails existing in POP3 server.

Syntax MessageCount ()

Return value : [out] The number of emails (VT_ I4)

Example

```
result = m_caoController.Execute("MessageCount")
```

3.1.2. CaoController::Execute(“ListUIDL”) command

Obtain the list of UIDL of emails.

Based on UIDLs obtained by this command, execute Size command, DeleteMessage command and CaoController::AddFile method.

Syntax ListUIDL()

Return value : [out] List of UIDL of mails (VT_ARRAY)
Obtain the list of UIDL of emails as VT_BSTR arrays.

Example

```
result = m_caoController.Execute("ListUIDL")
```

3.1.3. CaoController::Execute(“Size”) command

Obtain the size of a specified email by byte count.

Syntax Size (< UIDL >)

< UIDL >	:	[in] UIDL of an email (VT_BSTR)
Return value	:	[out] Size of an email (Byte) (VT_I4)

Example

```
result = m_caoController.Execute("UIDL", "TestUIDL")
```

3.1.4. CaoController::Execute(“DeleteMessage”) command

Delete a specified email from the server.

Syntax DeleteMessage (< UIDL >)

< UIDL >	:	[in] UIDL of a mail (VT_BSTR)

Example

```
m_caoController.Execute("DeleteMessage", "TestUIDL")
```

3.1.5. CaoController::Execute(“SetTimeout”) command

Set a timeout period by milliseconds.

Syntax SetTimeout (< Data >)

< Data >	:	[in] Timeout period (milliseconds) (VT_I4)

Example

```
m_caoController.Execute("SetTimeout", 10)
```

3.1.6. CaoController::Execute(“GetTimeout”) command

Obtain a timeout period by milliseconds.

Syntax GetTimeout ()

Return value : [out] Timeout period (milliseconds) (VT_I4)

Example

```
result = m_caoController.Execute("GetTimeout")
```

3.2. File class

Table 3-2 CaoFile::Execute command list

Command	Description	Page
GetHeaderNames	Obtain the list of header names.	P.10
GetHeader	Obtain a value corresponding to the specified header name.	P.11
GetBodyCount	Obtain the number of Body (part) of an email.	P.11
GetBodyFileName	Obtain a file name of the specified Body (part)	P.11
GetBody	Obtain the specified Body (part) number’s email body.	P.12
GetCodePage	Obtain a code page that encodes the specified Body.	P.12
GetLowMessage	Obtain a message of an email before analysis.	P.13

3.2.1. CaoFile::Execute(“GetHeaderName”) command

Obtain the list of header names.

Syntax GetHeadeName ()

Return value : [out]The list of header names. (VT_ARRAY)

Obtain the list of header names as arrays of VT_BSTR.

Example

```
result = m_caoFile.Execute("GetHeaderName")
```

3.2.2. CaoFile::Execute("GetHeader") command

Obtain a value corresponding to the specified header name.

If a header is encoded by MIME type, the decoded value will be returned.

Syntax GetHeader (< HeaderName >)

< HeaderName > : [in] Header name (VT_BSTR)

Return value : [out] The value of header corresponding to the specified header name (VT_BSTR | VT_ARRAY)

If the header name is[From], [To], [CC] or [Reply-To], multiple addresses that are identified by the corresponding values will be returned by VT_ARRAY.

For others, the corresponding value will be returned by VT_BSTR.

Example

```
result = m_caoFile.Execute("GetHeader", "HeaderName")
```

3.2.3. CaoFile::Execute("GetBodyCount") command

Obtain the number of Body (part) of an email.

If the target email consists of single part, the return value will be "1", if the target email consists of multiple parts, the return value will be "2" or higher.

Syntax GetBodyCount ()

Return value : [out] The number of Body of email (VT_I4)

Example

```
result = m_caoFile.Execute("GetBodyCount")
```

3.2.4. CaoFile::Execute("GetBodyFileName") command

Obtain a file name of the specified email's body number.

If the specified body is the main message of an email, null character will be returned as a file name.

If a file name is encoded by MIME or RFC2331 type, the decoded file name will be returned.

The available range of the number is from 0 to N (the number of body of email – 1).

Syntax GetBodyFileName (< No >)

< No > : [in] Email’s body number. (VT_I4)

Return value : [out] A file name of the email’s body. (VT_BSTR)
If the specified body number represents the main message of email, null character will be returned, and if it represents an attached file, a file name will be returned.

Example

```
result = m_caoFile.Execute("GetBodyFileName", 0)
```

3.2.5. CaoFile::Execute(“GetBody”) command

Obtain the specified body number’s email body with byte type array.

The available range of the number is from 0 to N (the number of body of email – 1).

Syntax GetBody (< No >)

< No > : [in] Email’s body number(VT_I4)

Return value : [out] Byte array of the main message of the email’s body (VT_ARRAY)

Example

```
result = m_caoFile.Execute("GetBody", 0)
```

3.2.6. CaoFile::Execute(“GetCodePage”) command

Obtain a code page that encodes the specified number’s email Body.

If an encode is not listed in Content-Type, 0 will be returned.

The available range of the number is from 0 to N (the number of body of email – 1).

Syntax GetCodePage (< No >)

< No > : [in] Email’s body number (VT_I4)

Return value : [out] A code page that encodes the specified Body (VT_I4)

Example

```
        result = m_caoFile.Execute("GetCodePage", 0)
```

3.2.7. CaoFile::Execute(“GetLowMessage”) command

Obtain a message of an email before analysis by byte array.

Syntax GetLowMessage ()

Return value : [out] Email message (VT_ARRAY)

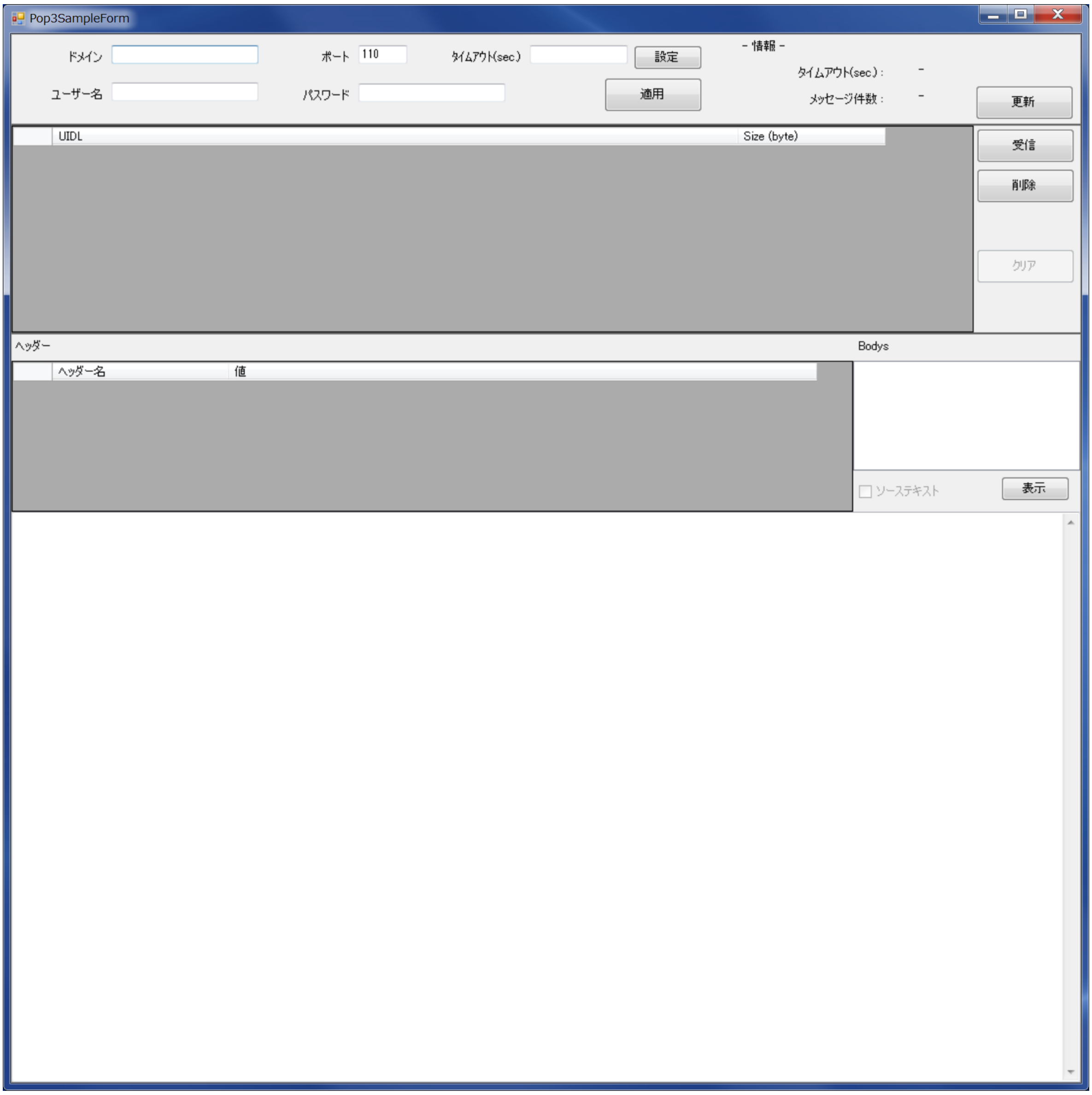
Example

```
        result = m_caoFile.Execute("GetLowMessage")
```

4. Sample program

This section introduces a sample program utilizing POP3 provider and Visual C#. This sample program is stored in the following folder.

List 4-1 **Pop3SampleForm.cs**



ポート 110

Figure 4-1 CAO Pop3 sample

Sample program

——————————————————————————————————————

‘ AddController

```
_ctrl = cCaoEngine.Workspaces[0].AddController("POP3", "CaoProv.POP3", string.Empty, optStr);

‘MessagesCount
int count = (int)_ctrl.Execute("MessageCount", null);

‘ ListUIDL
e.Result = _ctrl.Execute("ListUIDL", null);

‘ Size
int size = (int)_ctrl.Execute("Size", id);

‘ DeleteMessage
_ctrl.Execute("DeleteMessage", uidl);

‘ SetTimeout
_ctrl.Execute("SetTimeout", timeout);

‘ GetTimeout
int msec = (int)_ctrl.Execute("GetTimeout", null);

‘ ProviderCancel
_ctrl.Execute("ProviderCancel", null);

‘ ProviderClear
_ctrl.Execute("ProviderClear", null);

‘ AddFile
CCaoFile file = _ctrl.AddFile("mail", string.Format("UIDL={0}", uidl));

‘ Value
string[] texts = _file.Value as string[];

‘ GetHeaderNames
string[] headerNames = file.Execute("GetHeaderNames", null) as string[];

‘ GetHeader
object value = file.Execute("GetHeader", name);
```

```
‘ GetBodyCount
int count = (int)file.Execute("GetBodyCount", null);

‘ GetBodyFileName
string fileName = file.Execute("GetBodyFileName", i).ToString();

‘ GetCodePage
int codePage = (int)_file.Execute("GetCodePage", data.Num);

‘ GetLowMessage
byte[] mess = _file.Execute("GetLowMessage", null) as byte[];
```