

# Edgewall Trac providers

Version 1.0.0

User's Guide

July 2, 2019

Remark

**Revision history**

Version	Date	Content
1.0.0	2019-07-02	First edition

## 目次

1. Introduction .....	4
2. Provider overview .....	5
2.1. Operating system installation .....	5
2.2. Method Properties .....	6
2.2.1. CaoWorkspace::AddController method .....	6
2.2.2. CaoController::AddFile method.....	7
3. Command reference .....	8
3.1. Controller classes .....	8
3.1.1. CaoController::Execute("SetUserName ").....	9
3.1.2. CaoController::Execute("SetPassword ").....	9
3.2. File classes .....	10
3.2.1. CaoFile::Execute("Attach ").....	11
3.2.2. CaoFile::Execute("Create ").....	11
3.2.3. CaoFile::Execute ("Get") commands .....	12
3.2.4. CaoFile::Execute("Notify ").....	12
3.2.5. CaoFile::Execute ("Set") commands .....	13
3.2.6. CaoFile::Execute("Update ").....	13
4. Sample program .....	14
5. Appendix: .....	16
5.1. Appendix A. Elements of the ticket.....	16
5.2. Appendix B. XML-RPC.NET .....	16
5.3. Appendix C. THE MIT License .....	17

## 1. Introduction

Edgewall Trac providers use the XML-RPC library to provide a mechanism for accessing the Trac, such as ticketing the Trac and obtaining tickets.

For more information about Xml-RPC libraries, see:

This document provides an overview of Edgewall Trac providers and the CAO-interfaces (functional specifications) that are implemented.

[Related Site Links]

<http://www.xml-rpc.net>.

## 2. Provider overview

### 2.1. Operating system installation

The Edgewall Trac provider module consists of the following DLLs. If you install the module using the ORiN2 SDK installer, you do not need to install the module. If you are installing manually, perform the procedure as shown in Table 2-1.

RegistAsm.bat and UnregistAsm.bat are located in the DotNet ¥BAT folder below the ORiN2SDK installation folder.

Tabular 2-1 Edgewall Trac providers21

File name	CaoProvEdgewallTrac.dll
ProgID	CaoProv.Edgewall.Trac
Registry registration	RegistAsm.bat CaoProvEdgewallTrac.dll
Unregistering the registry	UnregistAsm.bat CaoProvEdgewallTrac.dll

## 2.2. Method Properties

### 2.2.1. CaoWorkspace::AddController method

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

For bstrOption, set the URL of the server. Use the XML-RPC. Specify the address with the/xmlrpc appended to the end of the URL on the top page of the Trac.

※ If the target servers use Trac Lightning, you must specify a/login/xmlrpc at the end to authenticate the logon.

< bstrCtrlName > : [in] controller name  
 < bstrProvName > : [in] Provider name. Fixed = "CaoProv.Edgewall.Trac"  
 <bstrPcName> : Running machine name of the in provider (unused)  
 <bstrOption> : In option string

You can specify the following options, separated by commas:

When using "=" or "" on the right side of the option, please put the parentheses on the right side.

Examples: PASSWORD=(pass=word)

URL : Specify the access point URL.  
Required

Examples: URL=  
http://localhost/redmine/

USERID : Specifies the userid to log in to the Trac.

If specified, it must be set to Password.

PASSWORD : Specifies the passwords for users who log in to the Trac.

If specified, it must be set to UserID.

Examples

UserID=admin>Password=password

#### **Example**

---

```
m_caoController = m_Workspace.AddController("ctrl", "CaoProv.Edgewall.Trac", "",
"URL=http://127.0.0.1/trac/project1/login/xmlrpc, USERID=admin, PASSWORD=password")
```

---

### 2.2.2. CaoController::AddFile method

**Form**      AddFile (<bstrName:BSTR>, <bstrOption:BSTR>)

Where bstrOption is the ticket-id of the Trac. If not specified, a new ticket is created.

< bstrName >            :   [in] file name  
<bstrOption>            :   In option string (ticket ID)

**Example**

---

```
m_caoController.AddFile("FileName", "ID=1")
```

---

## 3. Command reference

### 3.1. Controller classes

#### List of 3-1 CaoController::Execute Commands31

---

Commanded	Functioning	Page
SetUserName	Set a Username.	9
SetPassword	Set the password.	9

---

### 3.1.1. CaoController::Execute("SetUserName ")

Specifies the username for accessing the Trac.

**Form**      SetUserName (<Data>)

<Data>                    :    Username for accessing the Trac (VT\_BSTR)  
Returned value            :    Null

**Example**

---

```
m_caoController.Execute("SetUserName", "USER1")
```

---

### 3.1.2. CaoController::Execute("SetPassword ")

Specifies the passwords for users accessing the Trac.

**Form**      SetPassword (<Data>)

<Data>                    :    Passwords for Trac users (VT\_BSTR)  
Returned value            :    Null

**Example**

---

```
m_caoController.Execute("SetPassword", "PASSWORD1")
```

---

### 3.2. File classes

#### List of 3-2 CaoFile::Execute Commands<sup>32</sup>

---

Commanded	Functioning	Page
Attach	Sets an attachment.	11
Create	Creates a new ticket.	11
Get	Gets the elements of the specified ticket.	12
Notify	Sets whether to notify.	12
Set	Sets the elements of the specified ticket.	13
Update	Update the ticket.	13

### 3.2.1. CaoFile::Execute("Attach ")

Attaches a file to a ticket.

**Form** Attach (<Data>)

<Data> : [in] attachment information (VT\_VARIANT | VT\_ARRAY)

The breakdown is as follows.

1. File name (VT\_BSTR)
2. Description (VT\_BSTR (specify null if not set))
3. Attached file data (VT\_UI1|VT\_ARRAY)
4. Whether to overwrite a file with the same name if it exists (VT\_BOOL)

Returned value : Null

**Example**

---

```
m_caoFile.Execute("Attach", new object[] {"image.png", "my picture", data, true})
```

---

### 3.2.2. CaoFile::Execute("Create ")

Creates a new ticket.

At a minimum, "summary" and "description" must be set first with the "Set" commands.

※ For "summary" and "description", see "Ticket Element [\(Link\)](#)", and for ("Set") commands, see Section 3.2.5 [\(Link\)](#). [チケットの要素../01\\_input/Set - CaoFile::Execute\(](#)

**Form** Create ()

Returned value : Null

**Example**

---

```
m_caoFile.Execute("Create", null)
```

---

### 3.2.3. CaoFile::Execute ("Get") commands

Gets the elements for the specified track.

**Form**      Get (<Data>)

< Data >                :   [in] element name (VT\_BSTR)  
Returned value         :   Element contents (VT\_BSTR)

**Example**

---

```
Result = m_caoFile.Execute("Get", "summary")
```

---

### 3.2.4. CaoFile::Execute("Notify ")

Sets whether to notify. Set true to notify when updating tickets.

**Form**      Notify (<Data>)

< Data >                :   Presence/absence of [in] notification (VT\_BOOL)  
Returned value         :   Null

**Example**

---

```
m_caoFile.Execute("Notify", true)
```

---

### 3.2.5. CaoFile::Execute ("Set") commands

Sets the value for the specified element. Set in the form of "Element name = value"....

**Form** Set (<Data>)

< Data > : [in] Element name and value to set.  
Returned value : Null

**Example**

---

```
m_caoFile.Execute("Set", "owner=user1")
```

---

### 3.2.6. CaoFile::Execute("Update ")

Update the ticket. If you want to post a comment, specify the comment in the argument.

**Form** Update (<Data>)

< Data > : [in] comment (VT\_BSTR)  
Returned value : Null

**Example**

---

```
m_caoFile.Execute("Update", "Fixed. See also #1111")  
m_caoFile.Execute("Update")
```

---

## 4. Sample program

Here are some examples of programs using Trac providers and C#:

### Sample program

[Namespace WindowsFormsApplication2](#)

```
{
    Public partial class FrmMain : Form
    {
        Public FrmMain()
        {
            InitializeComponent();
        }

        Private void FrmMain_Load(object sender, EventArgs e)
        {
        }

        Private void buttonCreate_Click(object sender, EventArgs e)
        {
            Using (var engine = new CCaoEngine())
            {
                Var ws = engine.Workspaces[0];
                Var ctrl = ws.AddController("ctrls", "CaoProv.Edgewall.Trac", "", textBoxConnection.Text);
                Ctrl.Execute("SetUserName", textBoxUserId.Text);
                Ctrl.Execute("SetPassword", textBoxPassword.Text);
                Var file = ctrl.AddFile("AddFile", null);

                // Information
                File.Execute("Set", "owner="+textBoxOwner.Text);
                File.Execute("Set", "reporter="+textBoxReporter.Text);
                File.Execute("Set", "summary="+textBoxSummary.Text);
                File.Execute("Set", "description="+textBoxDescription.Text);
                File.Execute("Notify", false);

                // Create
                File.Execute("Create", null);
            }
        }
    }
}
```

```
// Attach a file
If (File.Exists(textBoxFilepath.Text))
{
    System.Drawing.Image img = System.Drawing.Image.FromFile(textBoxFilepath.Text);
    Byte[] imgData = ImageToByteArray(img);
    File.Execute("Attach", new object[] {textBoxFilename.Text, textBoxFileDescription.Text,
        ImgData, true});
}
Else
{
    Console.WriteLine("File does not exist.");
}
}
}

Public static byte[] ImageToByteArray(Image img)
{
    ImageConverter imgconv = new ImageConverter();
    Byte[] b = (byte[])imgconv.ConvertTo(img, typeof(byte[]));
    Return b;
}
}
}
```

## 5. Appendix:

### 5.1. Appendix A. Elements of the ticket

---

Element	Description
Owner	Representative
Reporter	Reporter
Cc	Participants
Summary	A summary (used as the title of a ticket)
Description	Advanced

### 5.2. Appendix B. XML-RPC.NET

XML-RPC.NET - XML-RPC for .NET

v2.5.0 Release

Copyright (C) 2001-2010 Charles Cook (chascook@gmail.com)

Xmlrpcgen

Copyright (C) 2003 Joe Bork

For more information about XML-RPC.NET visit <http://www.xml-rpc.net>.

XML-RPC.NET is licensed with MIT X11 license.

(see <http://www.xml-rpc.net/faq/xmlrpcnetfaq.html#6.12>)

For more information about XML-RPC refer to <http://www.xmlrpc.com/>

### **5.3. Appendix C. THE MIT License**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.