
記事

[Hiroshi Sato](#) · 2020年8月26日 4m read

Caché .Net BindingアプリケーションをIRISの.Net Native APIを利用して書き換える方法（その2）

ここで紹介するサンプルは、以下のGitHubから入手可能です。

[.NETサンプル](#)

Powershellから.NETインターフェースを呼び出すサンプル1

callclassmethod.ps1というファイル名です。

処理内容は、Caché（IRIS）のバージョン情報を表示するという非常に単純なものです。

Cachéでは.Net Bindingを使用していましたが、IRIS用に.Net Native APIで書き換えてみます。

以前のCachéのコードは以下の通りです。

```
Add-Type -Path 'C:/InterSystems/Ensemble/dev/dotnet/bin/v4.0.30319/InterSystems.Data.CacheClient.dll'  
$cc = New-Object InterSystems.Data.CacheClient.CacheConnection("localhost",1972,"USER","SYSTEM","SYS")  
$cc.Open()  
$ms = New-Object InterSystems.Data.CacheTypes.CacheMethodSignature  
$ms.Clear()  
$ms.SetReturnType($cc,[InterSystems.Data.CacheTypes.ClientTypeId]::tString)[InterSystems.Data.CacheTypes.C  
acheObject]::RunClassMethod($cc,"%SYSTEM.Version","GetVersion",$ms)  
$ms.ReturnValue.Value
```

IRISでは以下のように書き換えます。

```
Add-Type -Path 'C:/InterSystems/IRIS/dev/dotnet/bin/v4.5/InterSystems.Data.IRISClient.dll'  
$cc = New-Object InterSystems.Data.IRISClient.IRISConnection("localhost",1972,"USER","SYSTEM","SYS")  
$cc.Open()  
$iris = [InterSystems.Data.IRISClient.ADO.IRIS]::CreateIRIS($cc)  
$ReturnValue = $iris.ClassMethodString("%SYSTEM.Version","GetVersion")
```

\$ReturnValue

Powershell から .NET インタフェースを呼び出すサンプル2

classload.ps1 というファイル名です。

処理内容は、xml 形式のクラス定義を Caché /IRIS にロードする処理になります。

以前の Caché のコードは以下の通りです。

```
Add-Type -Path 'C:/InterSystems/Ensemble/dev/dotnet/bin/v4.0.30319/InterSystems.Data.CacheClient.dll'  
$cc = New-Object InterSystems.Data.CacheClient.CacheConnection("localhost",1972,"USER","SYSTEM","SYS")  
$cc.Open()  
$ms = New-Object InterSystems.Data.CacheTypes.CacheMethodSignature  
$ms.Clear()  
$ms.SetReturnType($cc,[InterSystems.Data.CacheTypes.ClientTypeId]::tString)  
$ms.Add("c:/temp/load.xml",1)  
[InterSystems.Data.CacheTypes.CacheObject]::RunClassMethod($cc,"%SYSTEM.OBJ","Load",$ms)  
$ms.ReturnValue.Value
```

IRIS では以下のように書き換えます。

```
Add-Type -Path 'C:/InterSystems/IRIS/dev/dotnet/bin/v4.5/InterSystems.Data.IRISClient.dll'  
$cc = New-Object InterSystems.Data.IRISClient.IRISConnection("localhost",1972,"USER","SYSTEM","SYS")  
$cc.Open()  
$iris = [InterSystems.Data.IRISClient.ADO.IRIS]::CreateIRIS($cc)  
$iris.ClassMethodStatusCode("%SYSTEM.OBJ","Load","c:/temp/load.xml","ck-d")
```

IRIS の方がコード量が少なくてすむことがわかりりいただけたと思います。

注意点は、そのクラスメソッドの戻り値のタイプによって、呼び出すメソッドが変わることです。

サンプル 1 の場合は、バージョン情報というテキスト情報が返ってくる想定ですので、 ClassMethodString を使用します。

サンプル 2 の場合は、戻り値はステータスコードが返ってくる想定ですので、 ClassMethodStatusCode を使用します。

.NET インタフェースを呼び出すコンソールアプリケーション

内容は、Powershellサンプルと同じです。

以下 Caché 用のコードです。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using InterSystems.Data.CacheClient;
using InterSystems.Data.CacheTypes;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            CacheConnection conn = new CacheConnection();
            conn.ConnectionString = "Server = localhost; Log File=cprovider.log;Port=1972; Namespace=Samples; Password = SYS; User ID = SYSTEM;";
            conn.Open();
            CacheMethodSignature ms = new CacheMethodSignature();
            ms.SetReturnType(conn, ClientType.tString);
            CacheObject.RunClassMethod(conn, "%SYSTEM.Version", "GetVersion", ms);
            Console.WriteLine("ReturnValue = " + ms.ReturnValue._Value);
            conn.Close();
        }
    }
}
```

IRISで .Net Native API を使用して書き換えたコードは以下になります。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using InterSystems.Data.IRISClient;
using InterSystems.Data.IRISClient.ADO;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            IRISConnection conn = new IRISConnection();
            IRIS iris;
```

```
conn.ConnectionString = "Server = localhost; Log File=irisnativeapi.log;Port=1972; Namespace=USER; Password = SYS; User ID = SYSTEM;";
conn.Open();
iris = IRIS.CreateIRIS(conn);
string ReturnValue = iris.ClassMethodString("%SYSTEM.Version", "GetVersion");
Console.WriteLine("ReturnValue = " + ReturnValue);
conn.Close();
}
}
}
```

[#InterSystems IRIS](#)

ソースURL:<https://jp.community.intersystems.com/post/cach%C3%A9-net-binding%E3%82%A2%E3%83%97%E3%83%AA%E3%82%B1%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%82%92iris%E3%81%AEnet-native-api%E3%82%92%E5%88%A9%E7%94%A8%E3%81%97%E3%81%A6%E6%9B%8E%E3%81%8D%E6%8F%9B%E3%81%88%E3%82%8B%E6%96%B9%E6%B3%95%EF%BC%88%E3%81%9D%E3%81%AE%EF%BC%92%EF%BC%89>