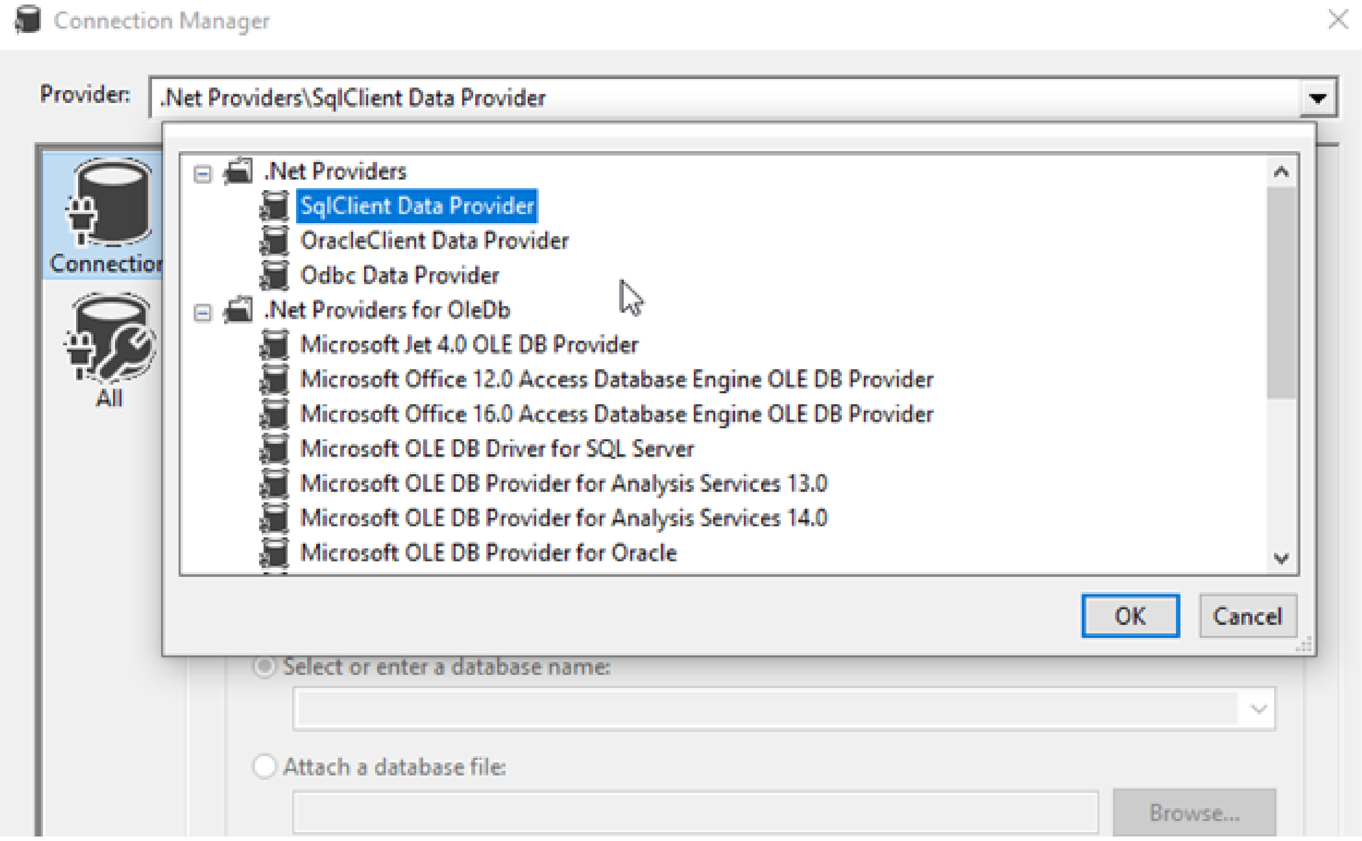
Article
Andreas Schneider · Mar 28, 2021 4m read

Registering the .Net Data Provider for InterSystems Caché and InterSystems IRIS as DbProviderFactories

This small tutorial described how to „register “ the ADO.NET Database Provider (Driver) for InterSystems IRIS and InterSystems Caché on a Windows machine.

Before we start: Why need the ADO.NET Database Provider to be registered?

ADO.NET provides factory classes standardize the way programmer will create a provider-specific Connection. This simplifies the way programmers create provider specific instances in a generic data access API. Details about that can be found here:
https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/obtaining-a-dbproviderfactory
If an Application use this system to obtain a list of available Data Providers, the provider needs to be registered. In my example I want to use the **.Net Data Provider for InterSystems IRIS** in a **MS SQL Server Integration Service Project**. This Application (SSIS) use a Connection Dialog that lists all on this machine available\known Data Provider:

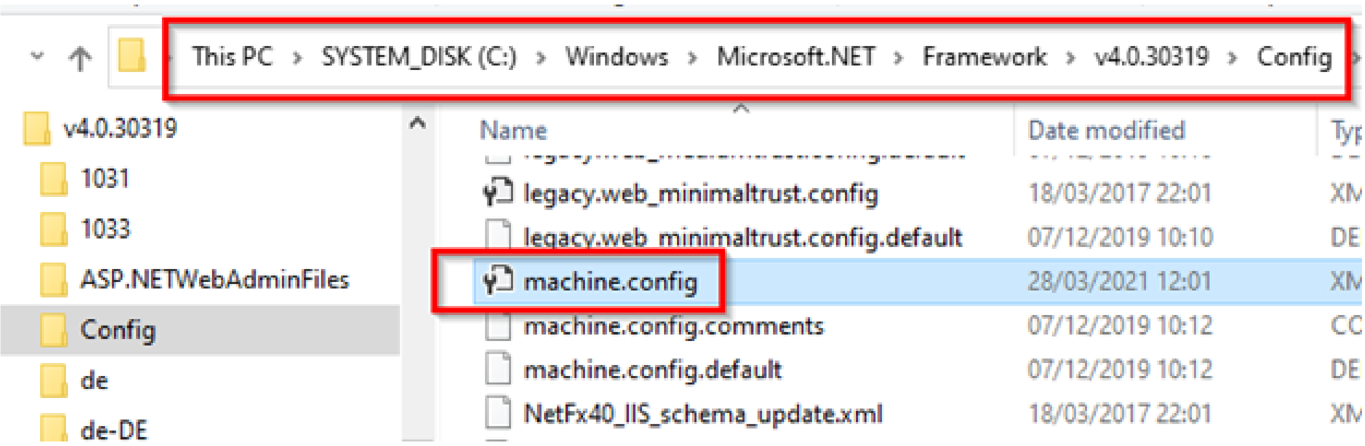

Connection Manager
Provider:
.Net Providers\SqlClient Data Provider
Connection
All
.Net Providers
SqlClient Data Provider
OracleClient Data Provider
Odbc Data Provider
.Net Providers for OleDb
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Provider
Microsoft Office 16.0 Access Database Engine OLE DB Provider
Microsoft OLE DB Driver for SQL Server
Microsoft OLE DB Provider for Analysis Services 13.0
Microsoft OLE DB Provider for Analysis Services 14.0
Microsoft OLE DB Provider for Oracle
OK
Cancel
Select or enter a database name:
Attach a database file:
Browse...

As you can see there are no InterSystems Provider available. The following section describes what needs to be done to list the provider on this connection dialog.

machine.config?!

Each .NET Framework data provider that supports a factory-based class registers configuration information in the DbProviderFactories section of the machine.config file on the local computer. The DbProviderFactories class uses the **machine.config** file(s) to resolve the invariant name to the appropriate provider factory type. The machine.config file is located in …\<windows>\Microsoft.NET\Framework\<version>\config\machine.config.

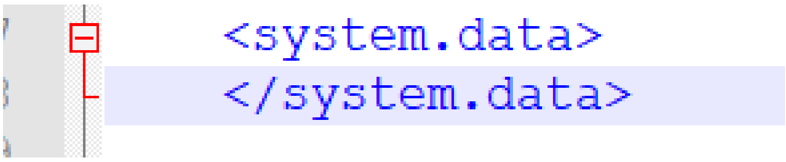
On my computer e.g:
C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config
and
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config


This PC › SYSTEM_DISK (C:) › Windows › Microsoft.NET › Framework › v4.0.30319 › Config
v4.0.30319
1031
1033
ASP.NETWebAdminFiles
Config
de
de-DE

Name	Date modified	Typ
legacy.web_minimaltrust.config	18/03/2017 22:01	XM
legacy.web_minimaltrust.config.default	07/12/2019 10:10	DE
machine.config	28/03/2021 12:01	XM
machine.config.comments	07/12/2019 10:12	CC
machine.config.default	07/12/2019 10:12	DE
NetFx40_IIS_schema_update.xml	18/03/2017 22:01	XM

Add DbProviderFactories

Open the machine.config file with a text editor and search for “ system.data ” . You found possibly an empty block like this:



```
    <system.data>
    </system.data>
```

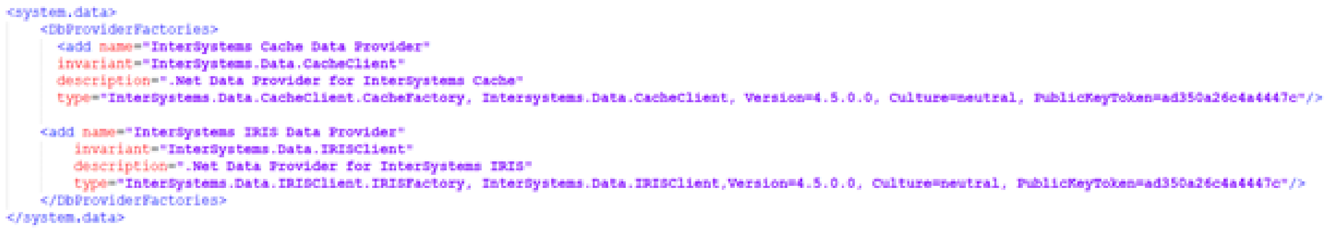
These are the values we have to insert for IRIS:

```
<add name="InterSystems IRIS Data Provider"
invariant="InterSystems.Data.IRISClient"
description=".Net Data Provider for InterSystems IRIS"
type="InterSystems.Data.IRISClient.IRISFactory,
InterSystems.Data.IRISClient,
Version=4.5.0.0,
Culture=neutral,
PublicKeyToken=ad350a26c4a4447c"/>
```

… and this for Caché

```
<add name="InterSystems Cache Data Provider"
invariant="InterSystems.Data.CacheClient"
description=".Net Data Provider for InterSystems Cache"
type="InterSystems.Data.CacheClient.CacheFactory,
Intersystems.Data.CacheClient,
Version=4.5.0.0,
Culture=neutral,
PublicKeyToken=ad350a26c4a4447c"/>
```

The “ system.block ” should now look like this:

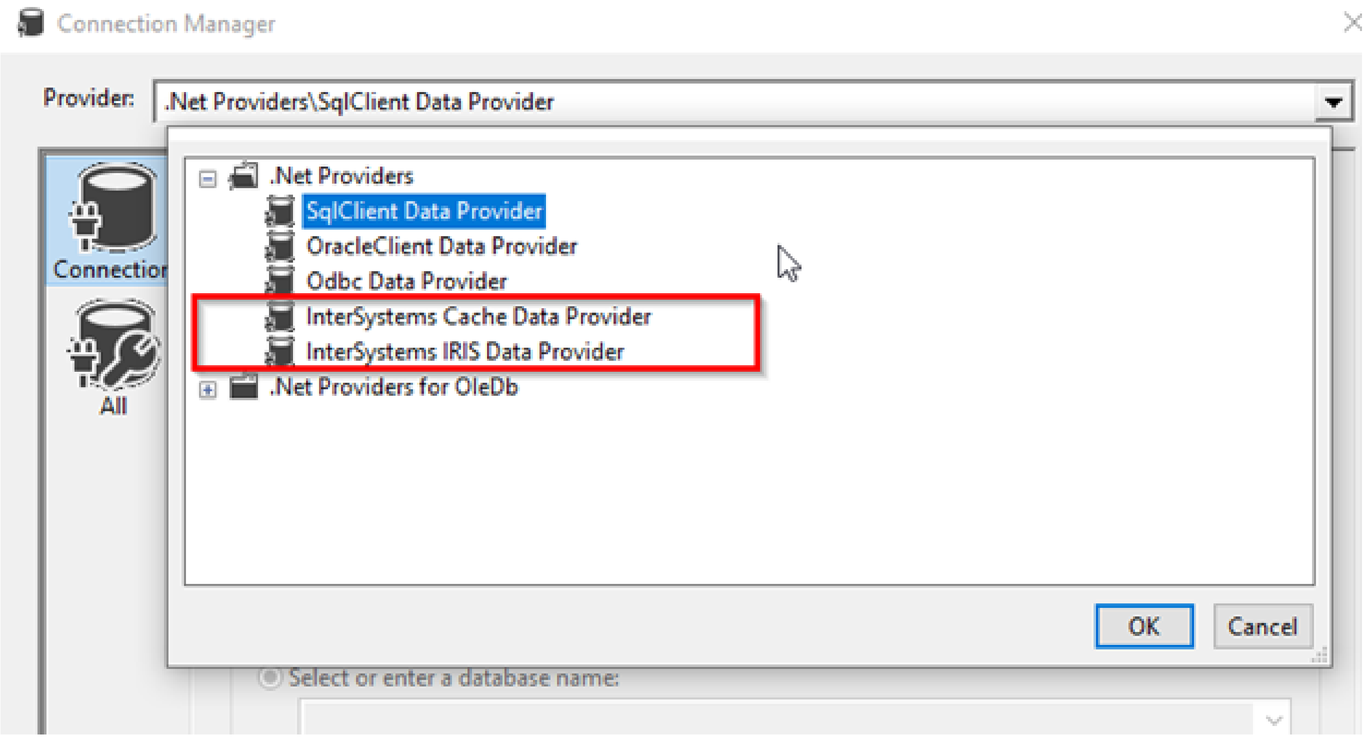


```
<system.data>
    <DbProviderFactories>
      <add name="InterSystems Cache Data Provider"
      invariant="InterSystems.Data.CacheClient"
      description=".Net Data Provider for InterSystems Cache"
      type="InterSystems.Data.CacheClient.CacheFactory, Intersystems.Data.CacheClient, Version=4.5.0.0, Culture=neutral, PublicKeyToken=ad350a26c4a4447c"/>

    <add name="InterSystems IRIS Data Provider"
        invariant="InterSystems.Data.IRISClient"
        description=".Net Data Provider for InterSystems IRIS"
        type="InterSystems.Data.IRISClient.IRISFactory, InterSystems.Data.IRISClient,Version=4.5.0.0, Culture=neutral, PublicKeyToken=ad350a26c4a4447c"/>
    </DbProviderFactories>
</system.data>
```

Pay attention to the <DbProviderFactories></DbProviderFactories> tags!

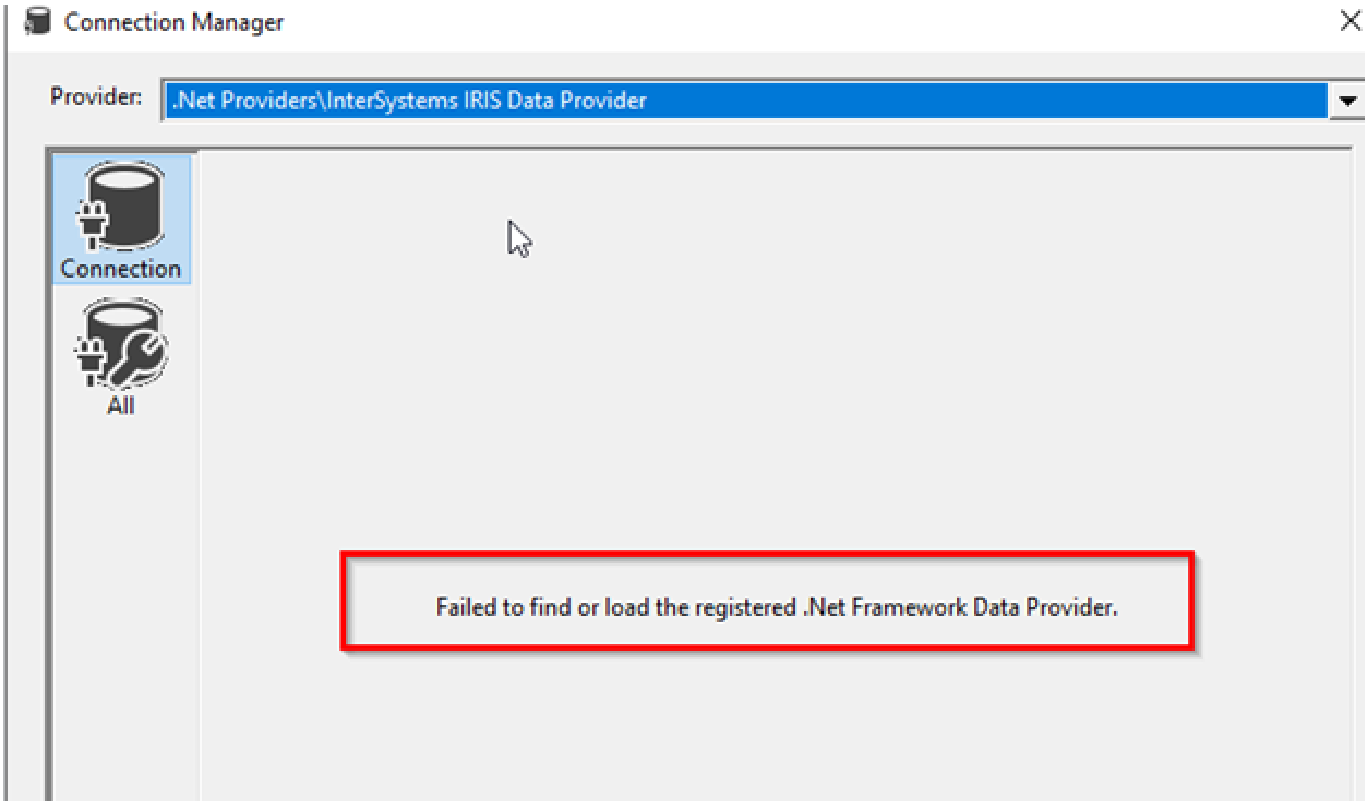
Now start Visual Studio (Data Tools), create a SSIS Projekt and open the Connection Manager again. The dialog should look like this:



Provider: .Net Providers\SqlClient Data Provider

Connection Manager
Connection
All
.Net Providers
SqlClient Data Provider
OracleClient Data Provider
Odbc Data Provider
InterSystems Cache Data Provider
InterSystems IRIS Data Provider
.Net Providers for OleDb
OK
Cancel
Select or enter a database name:

Great! The providers are listed!
... unfortunately that is not enough. The providers are not working if you select one of the new entries you get this message:



Provider: .Net Providers\InterSystems IRIS Data Provider

Failed to find or load the registered .Net Framework Data Provider.

That ’ s happened because the Provider are not known it the GAC (global assembly cache), so the entries can ’ t be found*.
*If you have done an IRIS \Caché installation on your computer (or just the client tools) maybe the provider is already registered in the GAC and you don ’ t this message and all is already working. Congratulations!

Register InterSystems provider in the GAC

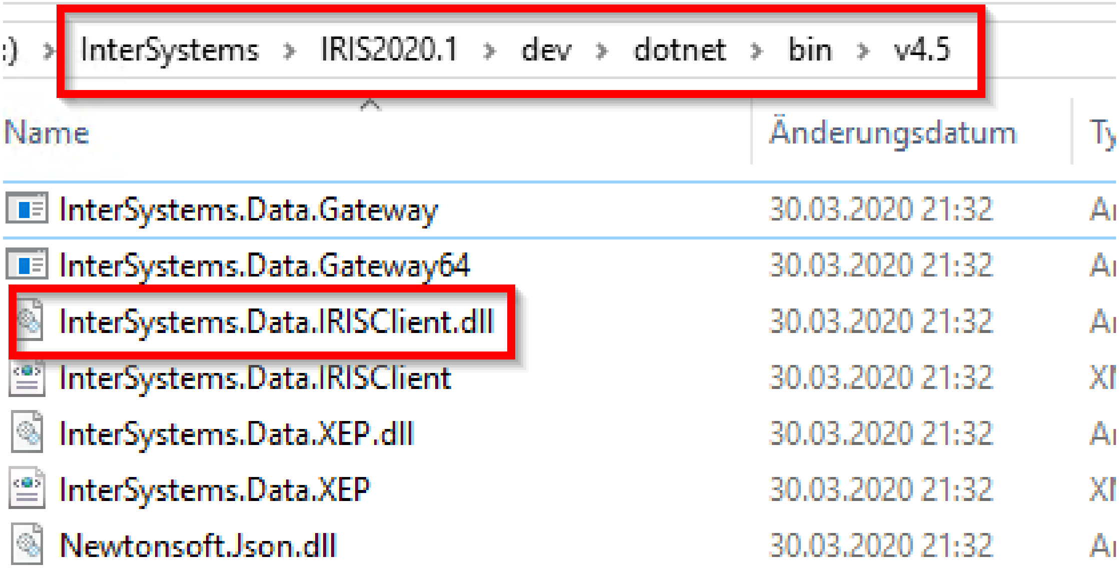
To registered the InterSystems provider on a computer you can do it this way:

Find the files first:

```
Intersystems.Data.CacheClient.dll (for Caché)

InterSystems.Data.IRISClient.dll (for IRIS)
```

Within an IRIS installation you can find the file e.g. …\dev\dotnet\bin\v4.5



:) › InterSystems › IRIS2020.1 › dev › dotnet › bin › v4.5

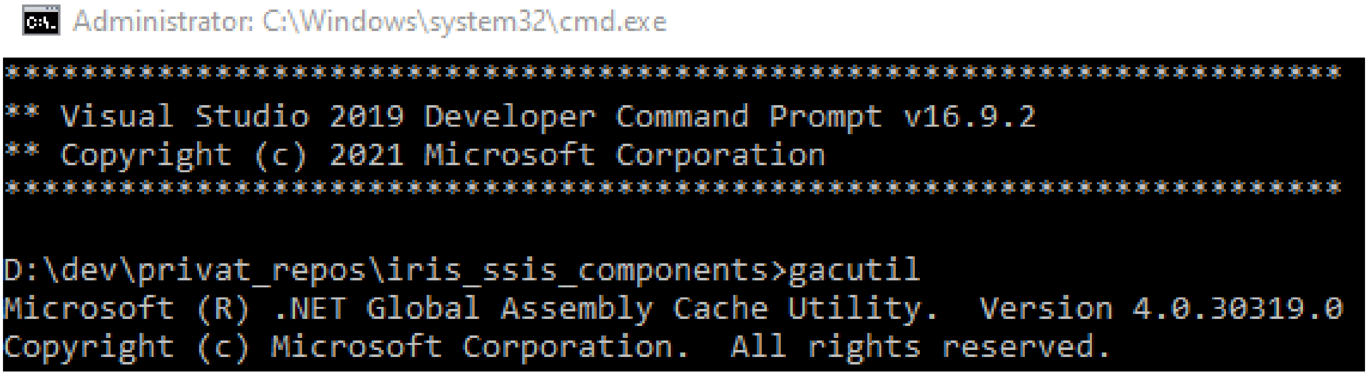
Name	Änderungsdatum	Ty
InterSystems.Data.Gateway	30.03.2020 21:32	A
InterSystems.Data.Gateway64	30.03.2020 21:32	A
InterSystems.Data.IRISClient.dll	30.03.2020 21:32	A
InterSystems.Data.IRISClient	30.03.2020 21:32	XI
InterSystems.Data.XEP.dll	30.03.2020 21:32	A
InterSystems.Data.XEP	30.03.2020 21:32	XI
Newtonsoft.Json.dll	30.03.2020 21:32	A

Copy the files on your computer in a folder (InterSystems.Data.IRISClient.dll).

Now you are able to register the Provider in the assembly cache with the gacutil.

Using gacutil

If you have a Visual Studio installed you can use the *Visual Studio Developer Command Prompt*:



Administrator: C:\Windows\system32\cmd.exe

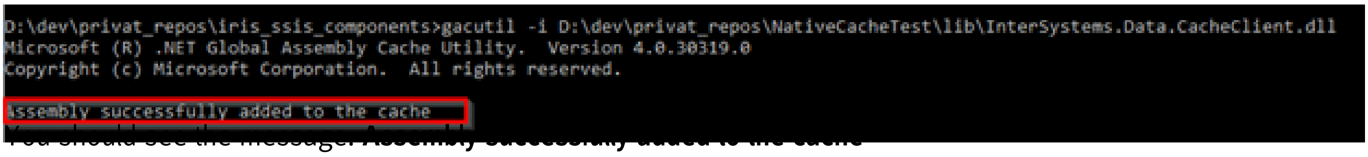
```
**********************************************************************
** Visual Studio 2019 Developer Command Prompt v16.9.2
** Copyright (c) 2021 Microsoft Corporation
**********************************************************************

D:\dev\privat_repos\iris_ssis_components>gacutil
Microsoft (R) .NET Global Assembly Cache Utility.  Version 4.0.30319.0
Copyright (c) Microsoft Corporation.  All rights reserved.
```

That has the advantage that you don ’ t need to look for the gacutil path, for installation just type:

```
gacutil -i <full path to your dll>\InterSystems.Data.CacheClient.dll

e.g.: gacutil -i D:\dev\privat_repos\NativeCacheTest\lib\InterSystems.Data.CacheClien
t.dll
```



```
D:\dev\privat_repos\iris_ssis_components>gacutil -i D:\dev\privat_repos\NativeCacheTest\lib\InterSystems.Data.CacheClient.dll
Microsoft (R) .NET Global Assembly Cache Utility.  Version 4.0.30319.0
Copyright (c) Microsoft Corporation.  All rights reserved.

Assembly successfully added to the cache
```

You should see the message: **Assembly successfully added to the cache**

To remove something from gac, use the option "-u": gacutil -u InterSystems.Data.CacheClient

Install both provider:

```
gacutil -i <full path to your dll>\InterSystems.Data.IRISClient.dll
gacutil -i <full path to your dll>\InterSystems.Data.CacheClient.dll
```

If you don ’ t have the “ Visual Studio Developer Command Prompt ” you need to search the gacutil executable first. On my computer it is located at: "C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\gacutil.exe"

Install:

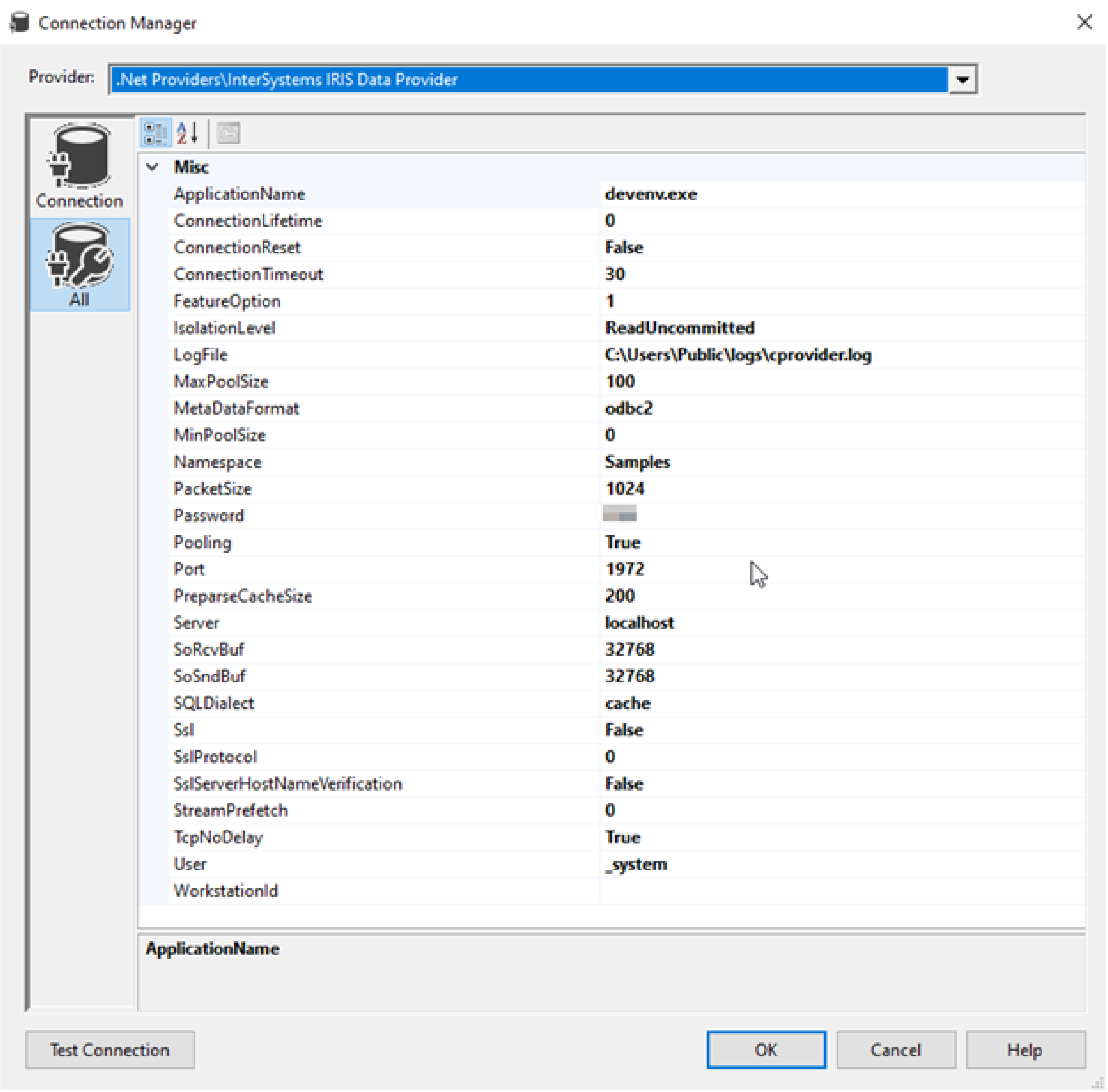
```
"C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\gacutil.exe
" -i D:\dev\privat_repos\NativeCacheTest\lib\InterSystems.Data.IRISClient.dll
"C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\gacutil.exe
" -i D:\dev\privat_repos\NativeCacheTest\lib\InterSystems.Data.CacheClient.dll
```

Remove:

```
"C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\gacutil.exe
" -u InterSystems.Data.CacheClient
"C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools\gacutil.exe
" -u InterSystems.Data.IRISClient
```

It works! :-)

And now… if you open the SSIS Project again, navigate to the Connection Manager it should looks like this:



Provider: .Net Providers\InterSystems IRIS Data Provider

Misc

ApplicationName	devenv.exe
ConnectionLifetime	0
ConnectionReset	False
ConnectionTimeout	30
FeatureOption	1
IsolationLevel	ReadUncommitted
LogFile	C:\Users\Public\logs\cprovider.log
MaxPoolSize	100
MetaDataFormat	odbc2
MinPoolSize	0
Namespace	Samples
PacketSize	1024
Password	[redacted]
Pooling	True
Port	1972
PreparseCacheSize	200
Server	localhost
SoRcvBuf	32768
SoSndBuf	32768
SQLDialect	cache
Ssl	False
SslProtocol	0
SslServerHostNameVerification	False
StreamPrefetch	0
TcpNoDelay	True
User	_system
WorkstationId	

ApplicationName

You can now select the IRIS provider, see and configure all available connections details.

Thanks for reading.

#Deployment #Development Environment #Caché #InterSystems IRIS

Source URL:https://community.intersystems.com/post/registering-net-data-provider-intersystems-cach%C3%A9-and-intersystems-iris-dbproviderfactories