



OpenEdge Data Provider for List & Label

Author: Thomas Wurl
Taste IT Consulting

Date: May 2014
Version: 1.0

Contents

- 1. Introduction 3**
- 2. Getting started 4**
 - 2.1. OpenEdge LL project..... 4
 - 2.2. First testing..... 5
 - 2.3. The Sports2000 sample application 5
- 3. Using the data provider in your application..... 10**
 - 3.1. Integrate the data provider 10
 - 3.2. Registering a table..... 11
 - 3.3. Adding sort orders for a table 11
 - 3.4. Adding table relations 11
- 4. Building the OpenEdge Adapter DLL 14**
- 5. License and Support..... 18**
- 6. Contact..... 19**

1. Introduction

This document describes Taste IT Consulting's Open Edge Data Provider for List & Label.

There are many ways to connect an OpenEdge application to List & Label, but the data provider approach is the most powerful. You just have to give List & Label access to your data, define your layouts and the List & Label engine does the rest.

Although an OpenEdge ProBindingSource can be used as a data source for List & Label, it doesn't give you all the features that a native data provider does.

The OpenEdge Data Provider for List & Label has two parts:

- The OpenEdgeAdapter DLL written in C#
- OpenEdge ABL classes on top of the DLL.

With this approach I was able to hide the entire data provider interface definitions including all of the .Net Generics stuff from OpenEdge. The result is a fast and stable data provider for both Open Edge 10 and Open Edge 11.

Enjoy!

2. Getting started

There are two ways to get the data provider:

- Download from Taste IT Consulting
- combit List & Label distribution.

Starting in summer 2014, combit will ship the data provider in the samples directory of List & Label. This includes the OpenEdge part and the DLL but not the source code of the DLL.

If you are interested in the source code or if you have the need to build your own DLL, you can download a full version at:

www.taste-consulting.de/downloads.

Since the version of List & Label being used to build the DLL has to match the version that is being used with OpenEdge, you may have to compile your own DLL if you use an older version of List & Label.

But now let's start.

2.1. OpenEdge LL project

Unpack the **OpenEdgeListLabel19.zip** and make sure that you have the following directory structure on your disk:

- C:\LLWS (OpenEdge workspace)
 - \LL (OpenEdge project)
- C:\Visual Studio (Visual Studio project).

Start the OpenEdge Architect (OpenEdge 10) / Developer Studio (OpenEdge 11) and choose C:\LLWS as workspace.

- Create a project called 'LL'
- Edit the 'LL' project properties
 - Choose 'Progress OpenEdge'
 - Edit 'Startup parameters' and add "-assemblies Assemblies"
 - Choose 'Database Connections' and connect to a Sports2000 database.

- Compile the 'ListLabel' directory.
- Done.

Make sure that everything compiles properly and that you don't see any compiler errors.

2.2. First testing

There are two samples in the ListLabel\Test directory:

- TestTempTable.p
- TestProDataset.p

Load either sample and try to run it. Examine the code to get an idea how it works.

2.3. The Sports2000 sample application

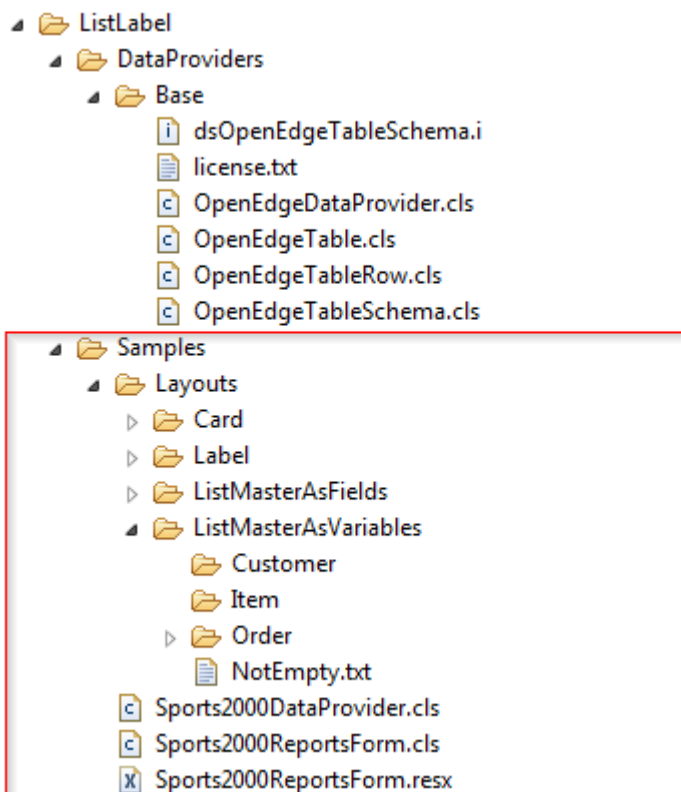
In the ListLabel\Samples directory you will find a small but powerful sample application that acts as a playground for the data provider. There are two classes:

- Sports2000DataProvider.cls
- Sports2000ReportForm.cls

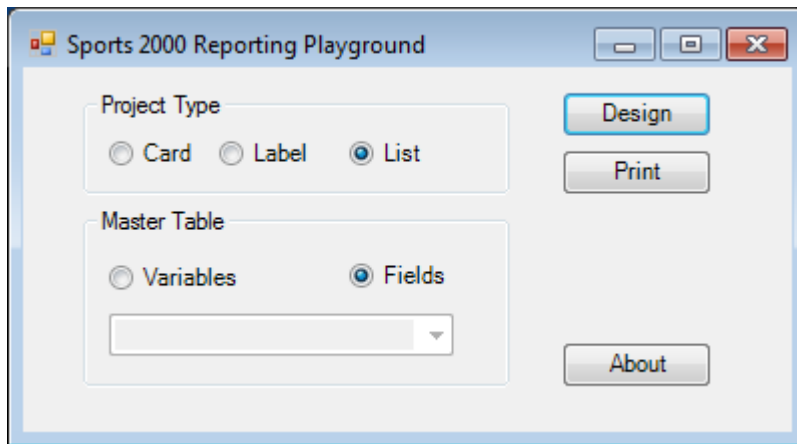
The Sports2000DataProvider is a subclass of the base data provider ListLabel.DataProviders.Base.OpenEdgeDataProvider.cls. It registers the entire Sports2000 database, including all relations and sort options for all fields. Examine the method 'registerDB' to see how it works.

The Sports2000DataProvider directly accesses the entire database without making use of temp-tables or prodatasets!

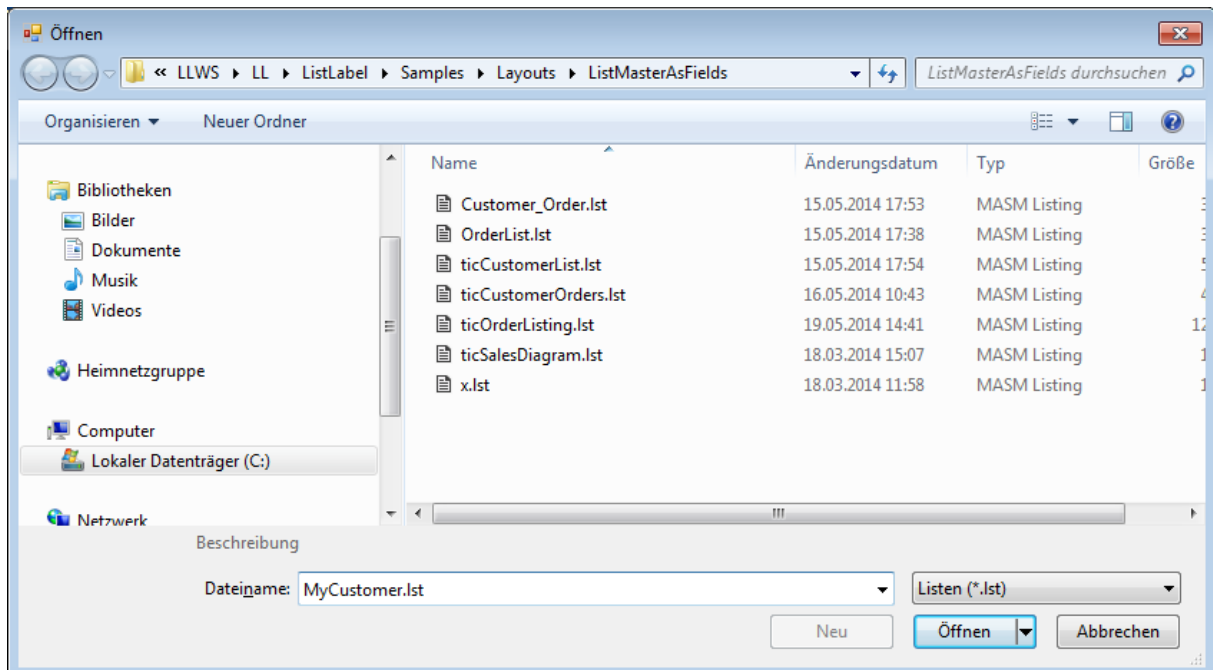
Load the Sports2000ReportsForm and run it. The program needs a special directory structure on the disk where the List & Label layouts are stored.



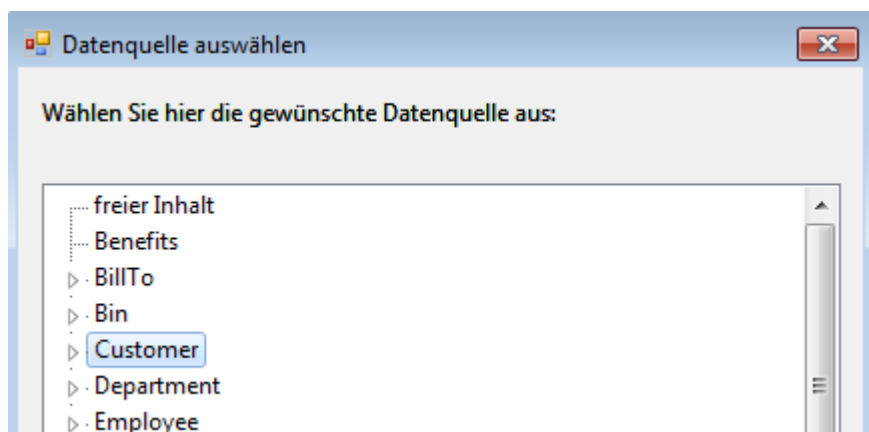
Getting started



Choose 'Design' to create your first layout and name it "MyCustomer.lst".

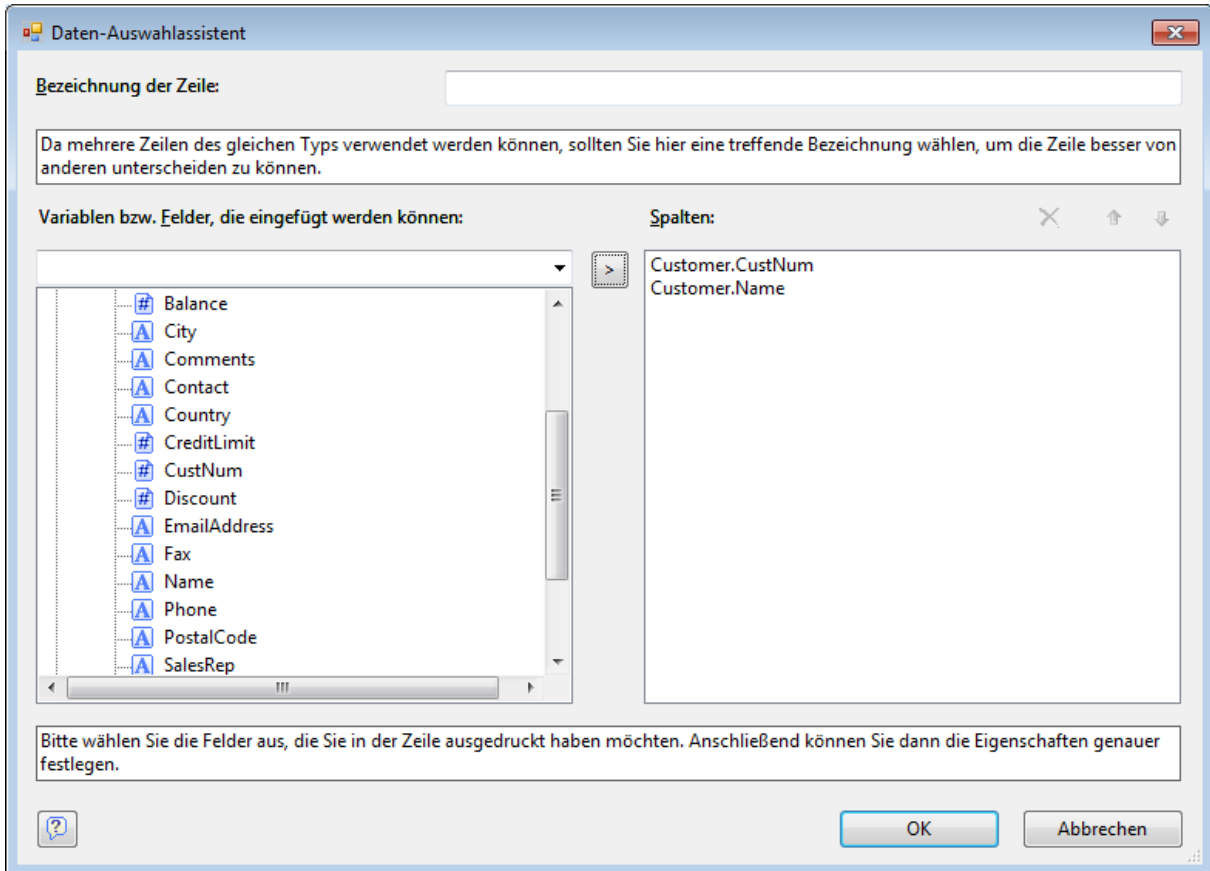


When the List & Label project wizard comes up just accept the defaults and create a table for "Customer".

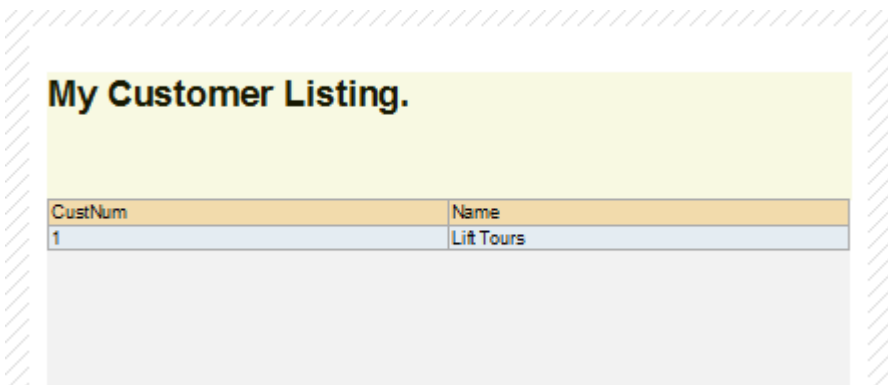


Getting started

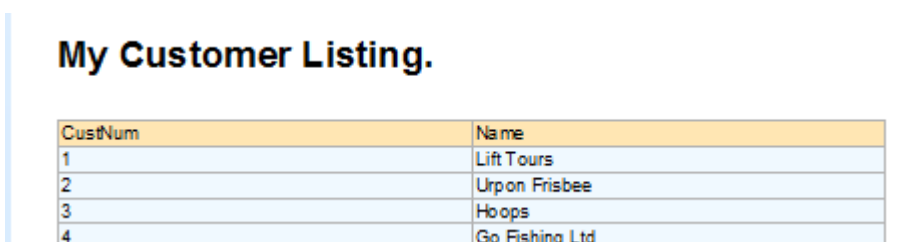
Choose some fields and press OK.



Update the title. Then you should have a layout like this.

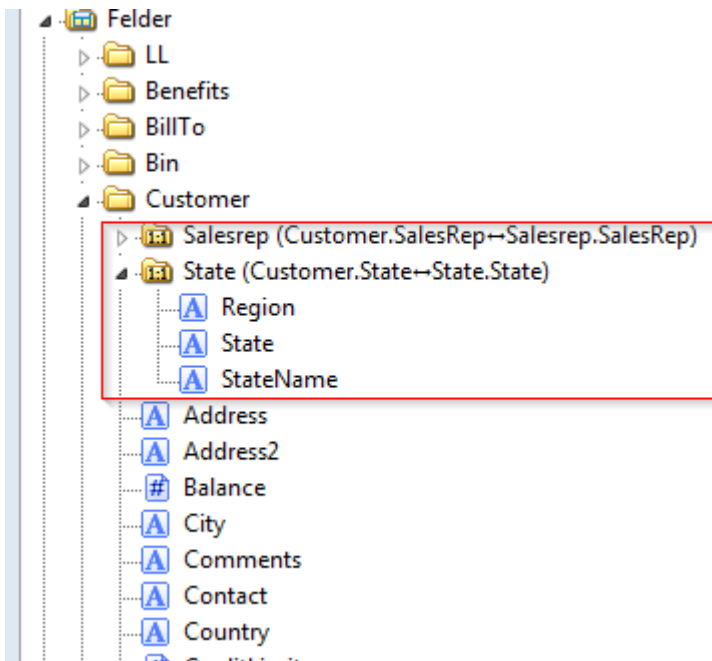


Save the layout, if you want to keep it. Choose the "Preview" tab run the report.

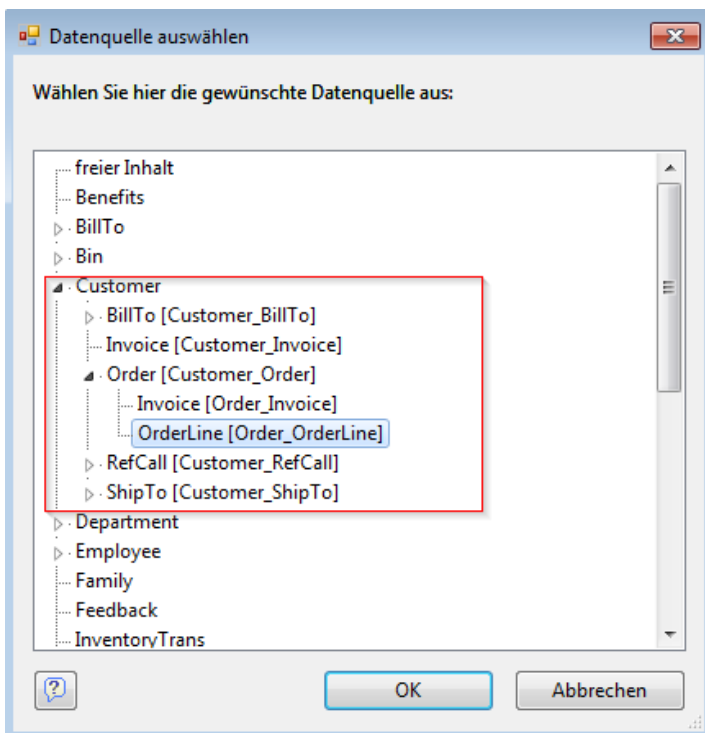


You are now done with your first data provider project!

Take a look at the Variable-/Fieldlist in the designer. Since the Sports2000DataProvider registers all relations of the Sports2000 database you have additional 1:1 relations under a table node giving you direct access to all foreign keys of a table.

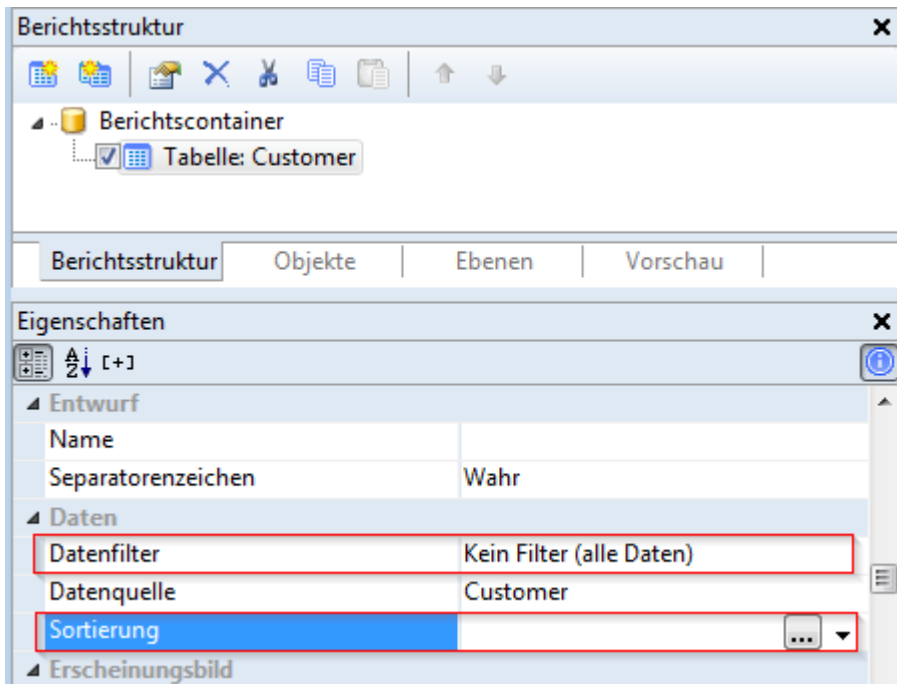


The same happens in the data source dialog. Here you see all child tables of a table.



Getting started

In the table properties you can add filters and sort options for all fields.



Take your time to try more complex things.

3. Using the data provider in your application

3.1. Integrate the data provider

If you want to integrate the data provider into your application, you just have to do the following.

Copy both DLL to your assembly directory

- combit.ListLabel19.dll
- TasteITConsulting.ListLabel19.OpenEdgeAdapter.dll
- Add assembly references for both dll.
- Copy the entire ListLabel ABL directory to your project. You may want to remove the samples and test directories.

Note: Make sure that the List & Label version in your assemblies matches the version the OpenEdgeAdapter has been built with. If the versions don't match you won't be able to compile your ABL code.

Here is a skeleton for your code:

```
USING Progress.Lang.*.
USING combit.ListLabel19.*.
USING ListLabel.DataProviders.Base.*.

/* Definitions */
DEFINE VARIABLE oLL           AS CLASS ListLabel NO-UNDO.
DEFINE VARIABLE oProvider     AS CLASS OpenEdgeDataProvider NO-UNDO.
DEFINE VARIABLE oTable       AS CLASS OpenEdgeTable NO-UNDO.

/* Creating instances */
oLL      = NEW ListLabel().
oProvider = NEW OpenEdgeDataProvider().

/* Bind */
oLL:ForceSingleThread = TRUE. /* required for OpenEdge :-(* */
oLL:DataSource = oProvider.

/* Define your data */
/* Put your fancy definitions here */

/* Action */
oLL:Design().

/* Cleanup */
oLL:dispose().
```

The data provider doesn't care where the data comes from. If you define an OpenEdgeTable, the table just receives a buffer handle. This buffer can be a temp-table buffer or a database buffer. In case of a temp-table buffer you have to make

sure that the data is already in the temp-table. Temp-tables can be used standalone or as part of a ProDataset.

ProDatasets are recommended when you get your data from an OpenEdge Appserver.

3.2. Registering a table

The table has two constructors. Use the second one, if you want to change the name for List & Label.

```
oTable = NEW OpenEdgeTable(phTableBuffer).
oTable = NEW OpenEdgeTable(phTableBuffer,pcTableName).

oTable = NEW OpenEdgeTable(TEMP-TABLE ttCustomer:DEFAULT-BUFFER-HANDLE,"Customer").
oProvider:addTable(oTable).
```

3.3. Adding sort orders for a table

The first parameter is the label shown in the designer, the second is the sort by phrase for the ABL query.

```
oTable:addSortOrder(pcSortDescription,pcSortBy).

oTable:addSortOrder("CustNum(+)", "BY CustNum").
oTable:addSortOrder("CustNum(-)", "BY CustNum DESCENDING").
```

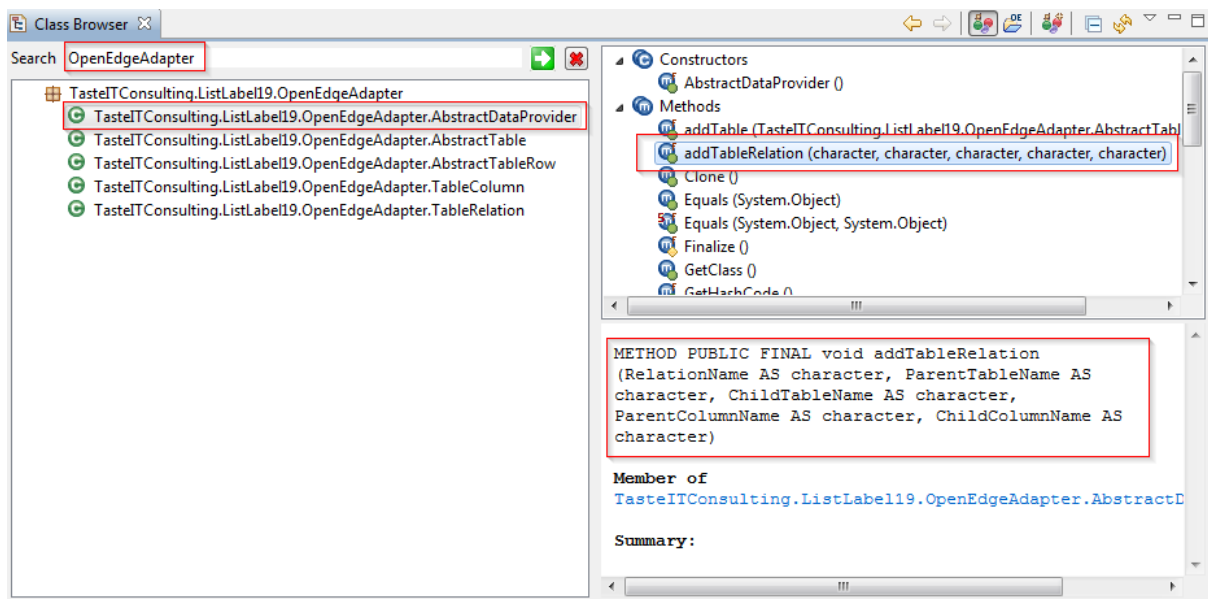
3.4. Adding table relations

```
oTable = NEW OpenEdgeTable(BUFFER Customer:HANDLE).
oProvider:addTable(oTable).
oTable = NEW OpenEdgeTable(BUFFER Order:HANDLE).
oProvider:addTable(oTable).

oProvider:addTableRelation("CustomerOrders","Customer","Order","CustNum","CustNum").
```

If the relation is based on multiple fields pass a field list to ParentColumnName and ChildColumnName. Use “,” or “~t” as a separator for the lists.

Table relations are defined in the provider. The OpenEdgeDataProvider class inherits most of its public methods from the DLL. List & Label table relations only exist in the DLL. Please use the class browser to get more knowledge.



Using ProDatasets

You can use your standard services like Business Entities or whatever to fill your datasets and pass it to the client. Filtering has to be built into your services if you just need a subset of your database data.

In case of ProDatasets the data provider doesn't access your database directly. It just takes what's inside a dataset.

To make your life a bit easier the data provider has support to register a dataset including its relations. There are two methods in the ABL OpenEdgeDataProvider to register a dataset:

```
/* 1) let the provider do the job */
hDataset = DATASET dsOrder:HANDLE.
oProvider:attachDataset(hDataset).

/* 2) define your own Table Schema, to be able to manipulate it */
oTableSchema = NEW OpenEdgeTableSchema(hDataset).
oTableSchema:addSortOrder (.....)
oProvider:attachDataset(hDataset,oTableSchema).
```

You can use the second version if you want to define additional SortOrders at the place where you collect your data.

Using the data provider in your application

```
/* TestProDataset.p */
DEFINE TEMP-TABLE eCustomer NO-UNDO LIKE Customer.
DEFINE TEMP-TABLE eOrder NO-UNDO LIKE Order.
DEFINE TEMP-TABLE eOrderLine NO-UNDO LIKE OrderLine.
DEFINE TEMP-TABLE eItem NO-UNDO LIKE Item.

DEFINE DATASET dsOrder FOR eOrder, eCustomer, eOrderLine, eItem
  DATA-RELATION FOR eOrder, eCustomer RELATION-FIELDS(CustNum, CustNum)
  DATA-RELATION FOR eOrder, eOrderLine RELATION-FIELDS (OrderNum,OrderNum)
  DATA-RELATION FOR eOrderLine, eItem RELATION-FIELDS (ItemNum, ItemNum).

/* Register */
hDataset = DATASET dsOrder:HANDLE.
oProvider:attachDataset(hDataset).
```

4. Building the OpenEdge Adapter DLL

Before you are able to use the data provider within your own application you may have to rebuild it.

There are different products and versions working together:

- List & Label version
- OpenEdge version, 32bit or 64bit
- Microsoft .NET Framework version

The latest, free version from the Taste IT Consulting always uses the newest release of List & Label and OpenEdge together with the Microsoft .NET Framework version that matches the OpenEdge version.

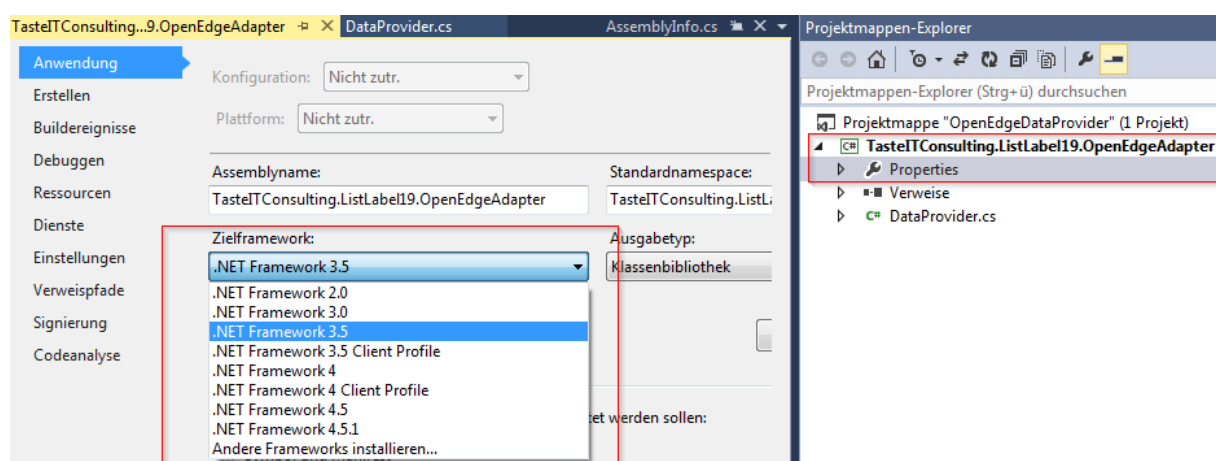
OpenEdge	Microsoft .NET
10.2B	3.5
11.X	4.0

The current release of List & Label (April 2014) is 19.002b.

To be able to edit and compile the C# code you need to have a Visual Studio installation. I recommend using **Visual Studio Express 2013 for Desktop**.

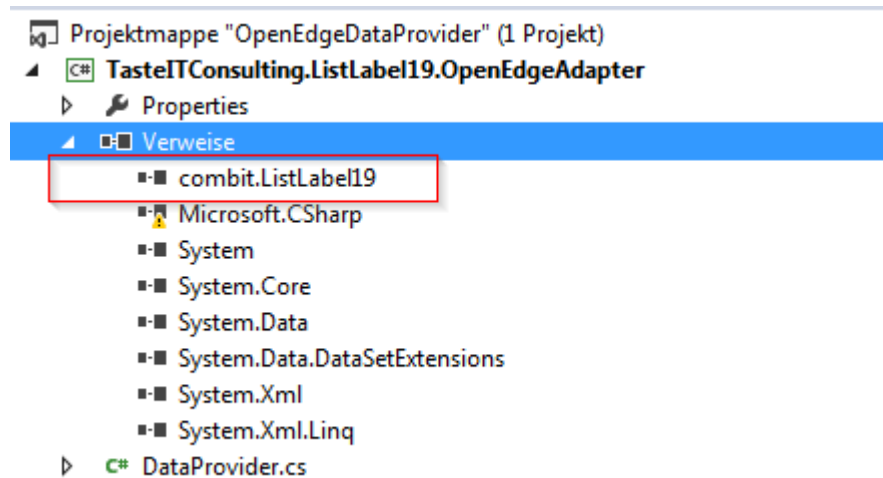
After you have installed VS ...

- Open ***OpenEdgeDataProvider.sln*** in Visual Studio.
- Edit Properties (Properties → right mouse click)
- Choose the Microsoft .NET Framework matching your requirements.



Building the OpenEdge Adapter DLL

- Add the proper List & Label DLL to the project.



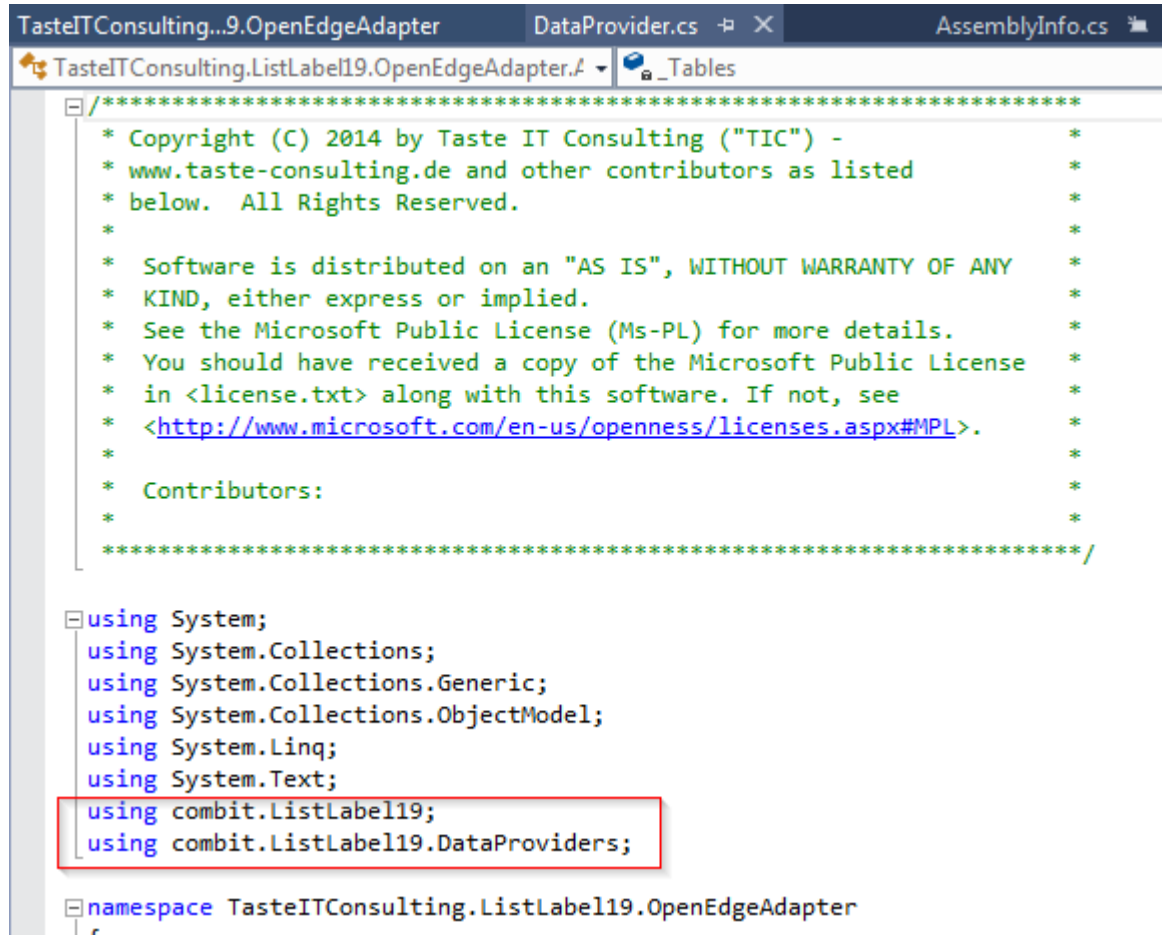
You will find the dll in your List & Label installation directory.

Note: Whenever you install a service pack from List & Label to have to replace this reference by the newest version and rebuild the dll. This also applies for following releases of List & Label.

I recommend creating a new project for each List & Label major version.

Building the OpenEdge Adapter DLL

Choose the right List & Label version in the c# DataProvider.cs



```

TasteITConsulting...9.OpenEdgeAdapter  DataProvider.cs  AssemblyInfo.cs
TasteITConsulting.ListLabel19.OpenEdgeAdapter.F  Tables

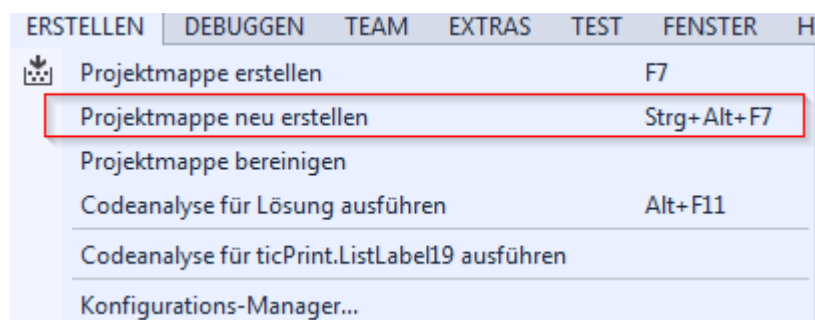
/*****
 * Copyright (C) 2014 by Taste IT Consulting ("TIC") -
 * www.taste-consulting.de and other contributors as listed
 * below. All Rights Reserved.
 *
 * Software is distributed on an "AS IS", WITHOUT WARRANTY OF ANY
 * KIND, either express or implied.
 * See the Microsoft Public License (Ms-PL) for more details.
 * You should have received a copy of the Microsoft Public License
 * in <license.txt> along with this software. If not, see
 * <http://www.microsoft.com/en-us/openness/licenses.aspx#MPL>.
 *
 * Contributors:
 *
 *****/

using System;
using System.Collections;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Text;
using combit.ListLabel19;
using combit.ListLabel19.DataProviders;

namespace TasteITConsulting.ListLabel19.OpenEdgeAdapter
{

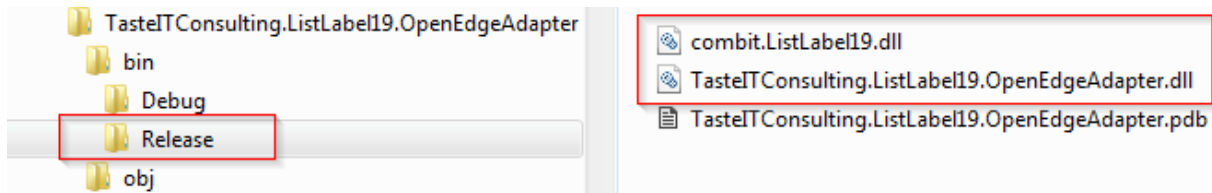
```

- Re-Build the Project.



Building the OpenEdge Adapter DLL

After you have done this, you will find both DLL in the release directory:



Copy both DLL to your ABL assembly directory, replace the references in assemblies.xml and recompile the ABL DataProvider and all of your own classes.

Congratulations! The c# part is done.

5. License and Support

The data provider is free of charge and Open Source. Please note that it is registered with the **MS-PL** license. Make sure to read **license.txt** and to accept it, before using the software.

I will do my very best to give the Progress Open Edge community a stable and powerful data provider. I'm also working closely together with combit to integrate new features early to be able to deploy a proper version with each new release of List & Label.

But there is no official support for the data provider!

Feel free to send me an email if you have any questions or trouble.

In the future I also may add a forum to my website or try to find a place in the progress communities where users can ask questions and share knowledge.

If you want more then please contact me. I'm able to offer additional support within consulting.

6. Contact

Taste IT Consulting
Thomas Wurl

Am Denkmal 13
31848 Bad Münde
Germany

thomas.wurl@taste-consulting.de
<http://www.taste-consulting.de>

...taste it!