

HCL Informix 15.0.0

Informix .NET EF Core Provider Reference Guide



Contents

- Chapter 1. Informix® .NET EF Core Provider Guide..... 3**
 - Overview of the HCL Informix® .NET EF Core Provider.....3
 - What is the Informix® .NET EF Core Provider?.....3
 - Supported programming environments.....3
 - Informix™ .NET EF Core Applications..... 3
 - Installing the Informix .NET EF Core Provider4
 - Using Informix .NET EF Core Provider.....4
 - Supported Data types..... 5
 - Example.....5
- Index.....9**

Chapter 1. Informix® .NET EF Core Provider Guide

These topics assume you are familiar with the Microsoft™ .NET EF Core specification and using HCL Informix® servers and databases.

Overview of the HCL Informix® .NET EF Core Provider

The topics in this overview describe the Informix .NET EF Core Provider and provide information on the environment in which to use it. These topics also provide installation and connection information and general information to help you get started using the Informix .NET EF Core Provider.

What is the Informix® .NET EF Core Provider?

The Informix .NET EF Core Provider or Informix .NET Entity Framework Core Provider is based on open source and cross platform version of Framework database access technology.

Informix .NET EF Core Provider is popular for the following reasons:

- Cross Platform application support
- Application migration is easy with very limited database provider specific code.

Supported programming environments

The Informix .NET EF Core Provider can be used by any application that can be run by the .NET Core.

The Informix .NET EF Core Provider is shipped with CSDK 15.0.0.0. It is available on Windows x64 and Linux x86_64.

Informix.EntityFramework.Core.dll is the name of Provider on both platforms and located at \$INFORMIXDIR\bin and is built on top of Informix .NET Core & ODBC Driver .

.NET EF Core Provider is built on following environment:

- Microsoft Windows Server 2016 Standard : v1607 Build 14393.2551
- .NET Core SDK Version : v3.1.410
- .NET EF Core version : v5.0.9
- Microsoft Visual Studio Enterprise 2017 : v15.8.6

Informix™ .NET EF Core Applications

The .NET Core applications using the Informix .NET EF Core Provider need other .NET Core assemblies/libraries from .NET Runtime. Such applications need to download/acquire the required assembly files (.NET Core SDK/Runtime).

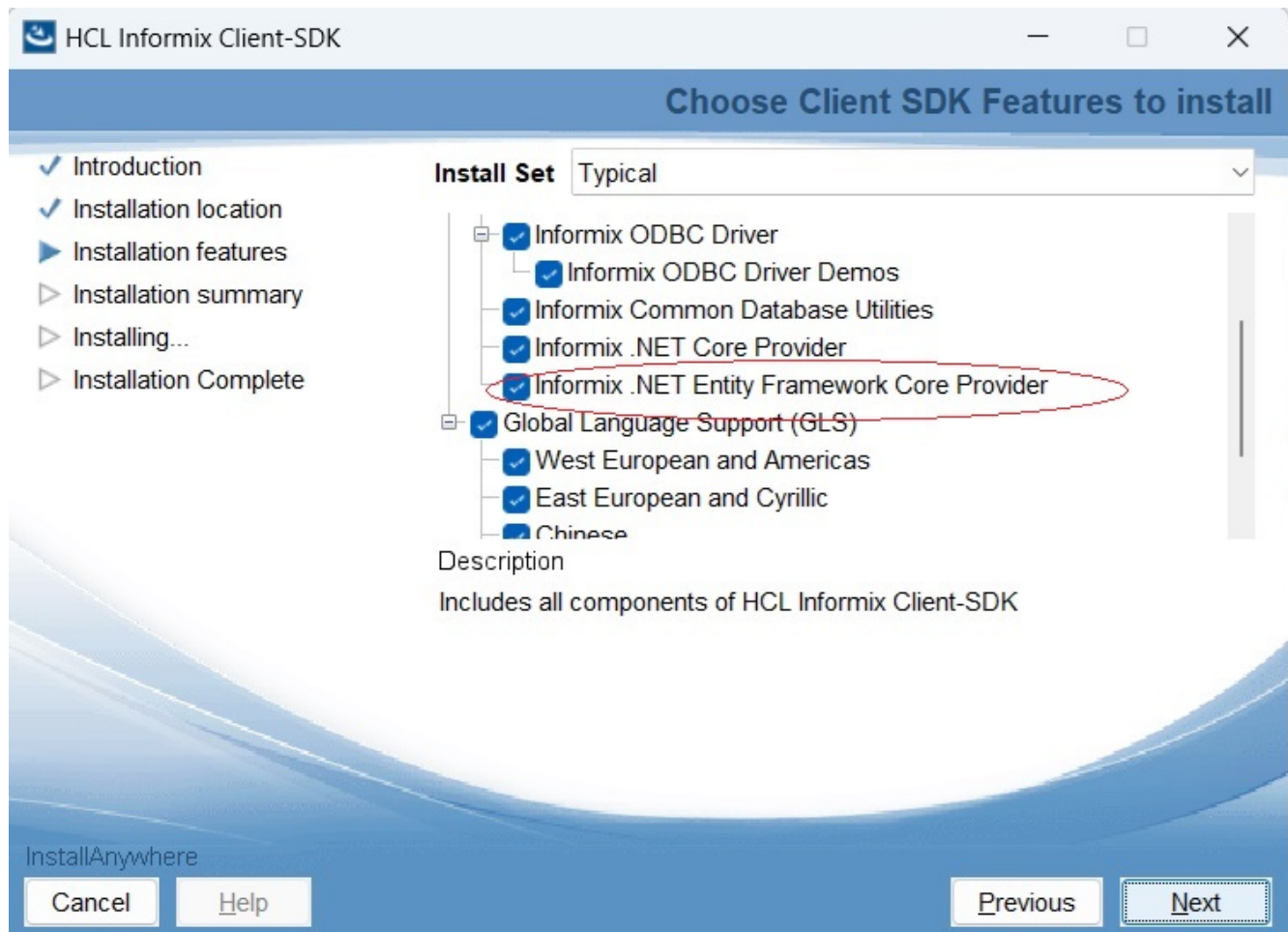
In order to run such applications, all the Informix related environment variables required i.e. INFORMIXDIR, INFORMIXSQLHOSTS, INFORMIXSERVER, PATH etc must be appropriately set.

For Sample program, see [Example on page 5](#).

Installing the Informix .NET EF Core Provider

You can install the Informix .NET EF Core Provider with the HCL® Informix® Client Software Development Kit (Client SDK) through a typical or custom installation.

When product is installed on Windows and Linux platforms, .NET EF core provider binary **Informix.EntityFramework.Core.dll** will be available in %INFORMIXDIR%\bin directory.



Using Informix .NET EF Core Provider

UseInformix(<Connection String>) is only override method specific to Informix, every .NET EF Core application calls method **OnConfiguring()**, in this method we need to call database provider specific connection API. Any database specific parameters needs to be passed as part of connection string.

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseInformix("Database=efcoredb; server=Informix; User ID=user1; Password=xxxx;
    connectdatabase=no");
}
```



Note: All the connection options are same as .NET Core Provider (ConnectionString property).

Supported Data types

Following data types are supported currently:

- INT
- CHAR, NCHAR
- VARCHAR, NVARCHAR
- DATETIME YEAR to MONTH
- DATE
- MONEY
- DECIMAL
- BIGINT
- SMALLINT
- SERIAL
- BIGSERIAL
- FLOAT
- SMALLFLOAT



Attention: Migration and Scaffolding operations: There are some limitations if tables has PK/FKs.

Example

```
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Migrations;
using Microsoft.EntityFrameworkCore.Storage;
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using static EFCoreSql.SchoolContext;

namespace EFCoreSql
{
    public partial class CreateDatabase : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.Sql("Some custom SQL statement");
            migrationBuilder.CreateTable(
                name: "Authors",
                columns: table => new
                {
                    AuthorId = table.Column<int>(nullable: false),
                    FirstName = table.Column<string>(nullable: true),
                    LastName = table.Column<string>(nullable: true)
                },
                constraints: table =>
```

```

        {
            table.PrimaryKey("PK_Authors", x => x.AuthorId);
        });
    }
}

class SchoolContext : DbContext
{
    public SchoolContext() : base()
    { }
    public class Student
    {
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int StudentId { get; set; }
        public string Name { get; set; }
    }
    public class Course
    {
        public int CourseId { get; set; }
        public string CourseName { get; set; }
    }
    public DbSet<Student> Students { get; set; }
    public DbSet<Course> Courses { get; set; }
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseInformix("Database=efcoredb; server=Informix; User ID=user1; Password=xxxx;
connectdatabase=no");
    }
}

class Program
{
    static void Main(string[] args)
    {
        int rowCount = 1;

        using (var schoolContext = new SchoolContext())
        {
            schoolContext.ChangeTracker.LazyLoadingEnabled = false;

            bool _canCo = schoolContext.Database.CanConnect();
            if (_canCo == true)
                Console.WriteLine("Informix database is good");
            else
                Console.WriteLine("Not able to connect to Informix database!!");

            bool _Created = schoolContext.Database.EnsureCreated();
            if (_Created == false)
                Console.WriteLine("Database exist in the Informix");
            else
                Console.WriteLine("Database either got created or became the current database " +
                    "since connection string is using connectdatabase=no, and tables got " +
                    "created if it didn't exist in Informix!!");

            List<Student> stds = new List<Student>()
            {
                new Student(){StudentId=1},
                new Student(){StudentId=2},
            }
        }
    }
}

```

```

        new Student(){StudentId=3}
    };

    Console.WriteLine("Deleting all records from students table...");
    schoolContext.Students.RemoveRange(stds);
    schoolContext.SaveChanges();

    Console.WriteLine("Selecting record(s) from students table...");
    var _students = schoolContext.Students.FromSqlRaw("Select * from students;");
    var counter = _students.GetEnumerator();
    bool hasRow = counter.MoveNext();
    if (hasRow is false)
        Console.WriteLine("No records found - Expected");
    else
    {
        Console.WriteLine("Records found - NOT Expected!!");
        rowCount = 1;
        foreach (var row in _students)
        {
            Console.WriteLine("Row #" + rowCount + " = StudentId : " + row.StudentId
                + ", Name : " + row.Name);
            rowCount++;
        }
    }

    Console.WriteLine("Inserting 3 records into students table...");
    // Insert records into Students table.
    var std1 = new Student() { StudentId = 1, Name = "Sheshnarayan" };
    var std2 = new Student() { StudentId = 2, Name = "Vardaan" };
    var std3 = new Student() { StudentId = 3, Name = "Samriddhi" };

    schoolContext.Students.AddRange(std1, std2, std3);
    schoolContext.SaveChanges();

    Console.WriteLine("Selecting record(s) from students table...");
    _students = schoolContext.Students.FromSqlRaw("Select * from students;");
    counter = _students.GetEnumerator();
    hasRow = counter.MoveNext();
    if (hasRow is false)
        Console.WriteLine("No records found - NOT Expected!!");
    else
    {
        Console.WriteLine("Records found - Expected");
        rowCount = 1;
        foreach (var row in _students)
        {
            Console.WriteLine("Row #" + rowCount + " = StudentId : " + row.StudentId + ", Name : "
+ row.Name);

            rowCount++;
        }
    }
}

using (var updateschoolContext = new SchoolContext())
{
    updateschoolContext.ChangeTracker.LazyLoadingEnabled = false;

    bool _Created = updateschoolContext.Database.EnsureCreated();

```


Index

Special Characters

.NET EF Core Provider Guide 3

A

ASP.NET 3

C

Client SDK 3

H

HCL

Informix

.NET EF Provider

installing 4

HCL

Informix

Client SDK

3

HCL

Informix

ODBC Driver

3

HCL

Informix

OLE DB Provider

3

I

Informix

Client SDK

3

Informix

ODBC Driver

3

Informix

OLE DB Provider

3

Installing

HCL

Informix

.NET EF Provider

4

M

Microsoft .NET Framework SDK 3

Microsoft ODBC .NET EF 3

Microsoft OLE DB .NET EF 3

O

ODBC .NET 3

ODBC Driver 3

OLE DB .NET 3

OLE DB Provider 3

P

Platforms 3

V

Visual BASIC .NET 3

Visual C# .NET 3

Visual J# .NET 3