# FICO® Xpress

8.8

.NET INTERFACE

FICO® Xpress Optimization

FICO® Xpress 8.8

Deliverable Version: A

Last Revised: November 21, 2019

# Contents

CHAPTER 1

# Xpress Optimizer .NET Interface

Applications written in a .NET language, such as Visual C#, Visual Basic, or Visual C ++, can use the Xpress Optimizer via the .NET wrapper library.

## Introduction

The best way to get started using the wrapper is to open up the example projects included with this package and experiment with them.

The auto-completion feature in Visual Studio can be used to obtain a full list of class methods and properties and prototypes of each method.

The Optimizer .NET Interface can be used with .NET Framework version 3.5 and greater, and .NET Core 3.0 and greater.

## Using the Wrapper in Projects

The Optimizer functionality is exposed through the `XPRS` and `XPRSprob` classes, which reside in the Optimizer namespace, which resides in the `xprsdn.dll` assembly.

Each source file that uses the Optimizer should import the Optimizer namespace and should be compiled with a reference to `xprsdn.dll`.

To add a reference to a Visual Studio project, select **Add Reference** from the **Project** menu. Click the **Browse** button and locate `xprsdn.dll` in the `XpressMP\bin` folder.

Users of the .NET command line compilers, for example, `csc` for C#, or `vbc` for Visual Basic, can add a reference with the `/reference` option:

```
csc /reference:<path to xprsdn.dll> <your source file>
```

# Using the Interface

The .NET wrapper for the Xpress Optimizer has been designed to look and feel like the Common Language Runtime.

Compare code using the existing C Optimizer interface with the following C# example:

```
using Optimizer;
XPRS.Init("");
Console.WriteLine(XPRS.GetBanner());
XPRSprob prob = new XPRSprob();
prob.ReadProb("myprob","");
prob.Maxim("g");
prob.Destroy();
XPRS.Free();
```

The Optimizer functions have been renamed and put into classes. The function name is missing the XPRS prefix, and according to the .NET naming convention, each word is capitalized. Functions operating on a problem pointer are methods of an `XPRSprob` object; the rest are static methods of the XPRS class.

An example of usage for this function, as demonstrated in the FICO Xpress Optimization Examples Repository:

```
public void OptimizerMsg (XPRSprob prob, object data, string message, int len, int
msglvl)
        {
            switch(msglvl)
            {
                case 3:
                case 4:
                    Console.WriteLine ("{0}" + message, data);
                    break;
            }
        }
```

# Controls and Attributes

The Optimizer controls and attributes are properties of the `XPRSprob` object.

They have been renamed in a similar way to the functions:

```
prob.Presolve = 0;
```

The `MIPStatus` and `LPStatus` attributes are no longer integers but have enumeration types, `MIPStatus` and `LPStatus`:

```
if (prob.MIPStatus == MIPStatus.Infeasible) {
...
```

# Callbacks

The Optimizer callbacks are accessed using the .NET event / delegate mechanism.

For example, to install a method `OptimizerMsg` of the current class as a callback to receive Optimizer messages, you must to create a `MessageCallback` delegate object from the function and add it to the `MessageCallbacks` event field of the problem:

```
prob.MessageCallbacks += new MessageCallback(this.OptimizerMsg);
```

To remove a callback you must still create a delegate object, but use the `-=` operator to remove it from the event field:

```
prob.MessageCallbacks -= new MessageCallback(this.OptimizerMsg);
```

# Return Codes and Error Checking

Most of the functions in the C interface to the Optimizer use an integer return code to indicate if an error occurred.

The `XPRS` and `XPRSprob` member functions instead use the .NET exception mechanism to handle errors, throwing an exception of type `XPRSException`. This makes error handling easier, since a check is not required with every function call:

```
try {
  XPRS.Init("");
  Console.WriteLine(XPRS.GetBanner());
  XPRSprob prob = new XPRSprob();
  prob.ReadProb("myprob","");
  prob.MipOptimize("g");
  } catch (XPRSException e) {
  Console.WriteLine (e);
}
```

Because the return code is not used to indicate error status, several methods instead return data that in the C interface is returned via an output parameter. In general, a function that retrieves a single value returns it, whereas a function that retrieves more than one value is declared void and uses output parameters. For example:

```
/* In C */
char name[32];
int ret = XPRSgetprobname (name);
if (ret != 0) {
...
/* In C# */
try {
string name = prob.GetProbName ();
} catch (XPRSException e) {
...
```

# Destroying the Problem

Since .NET uses a garbage collector, it does not provide a mechanism for explicitly destroying objects.

Optimizer problems can take up large amounts of memory and so a `Destroy` function is provided to free up this memory. Any operation performed on an `XPRSprob` object after it has been destroyed will throw an exception.

```
XPRSprob prob = new XPRSprob ();
...
prob.Destroy ();
```

# Branch Bounds

Another notable change with the .NET interface is to the functions used to store and set the bounds for node separation during a global search.

The C interface function `XPRSstorebounds` returns a void pointer, representing a set of bounds, to the user, which is then passed into the call to `XPRSsetbranchbounds`. The .NET interface replaces the void pointer with a `Bounds` object, as demonstrated in the **setbranchbounds** example and below:

```
Bounds b = prob.StoreBounds (nbnds, mcols, qbtype, dbd);
prob.SetBranchBounds (b);
```

APPENDIX A

# Contacting FICO

FICO provides clients with support and services for all our products. Refer to the following sections for more information.

## Product Support

FICO offers technical support and services ranging from self-help tools to direct assistance with a FICO technical support engineer. Support is available to all clients who have purchased a FICO product and have an active support or maintenance contract. You can find support contact information and a link to the Customer Self Service Portal (online support) on the Product Support home page (www.fico.com/en/product-support).

The FICO Customer Self Service Portal is a secure web portal that is available 24 hours a day, 7 days a week from the Product Support home page. The portal allows you to open, review, update, and close cases, as well as find solutions to common problems in the FICO Knowledge Base.

## Product Education

FICO Product Education is the principal provider of product training for our clients and partners. Product Education offers instructor-led classroom courses, web-based training, seminars, and training tools for both new user enablement and ongoing performance support. For additional information, visit the Product Education home page (www.fico.com/en/product-training) or email producteducation@fico.com.

## Product Documentation

FICO continually looks for new ways to improve and enhance the value of the products and services we provide. If you have comments or suggestions regarding how we can improve this documentation, let us know by sending your suggestions to techpubs@fico.com.

Please include your contact information (name, company, email address, and optionally, your phone number) so we may reach you if we have questions.

# Sales and Maintenance

If you need information on other Xpress Optimization products, or you need to discuss maintenance contracts or other sales-related items, contact FICO by:

- Phone: +1 (408) 535-1500 or +44 207 940 8718
- Web: http://www.fico.com/en/analytics/optimization and use the available contact forms

# Related Services

**Strategy Consulting**: Included in your contract with FICO may be a specified amount of consulting time to assist you in using FICO Xpress to meet your business needs. Additional consulting time can be arranged by contract.

**Conferences and Seminars**: FICO offers conferences and seminars on our products and services. For announcements concerning these events, go to www.fico.com or contact your FICO account representative.

# FICO Community

The FICO Community is a great resource to find the experts and information you need to collaborate, support your business, and solve common business challenges. You can get informal technical support, build relationships with local and remote professionals, and improve your business practices. For additional information, visit the FICO Community (community.fico.com/welcome).

# About FICO

FICO (NYSE:FICO) powers decisions that help people and businesses around the world prosper. Founded in 1956 and based in Silicon Valley, the company is a pioneer in the use of predictive analytics and data science to improve operational decisions. FICO holds more than 165 US and foreign patents on technologies that increase profitability, customer satisfaction, and growth for businesses in financial services, telecommunications, health care, retail, and many other industries. Using FICO solutions, businesses in more than 100 countries do everything from protecting 2.6 billion payment cards from fraud, to helping people get credit, to ensuring that millions of airplanes and rental cars are in the right place at the right time. Learn more at www.fico.com.

# Index