

Forum:[Hack Forums](#)

Thread: [A simple customizable licensing system](#)

itcorp - November 25, 2018, 1:15 pm

This license serial manager is flexible and simple, it's not a professional solution so do not be too rough in commenting.

Example of usage with base class with default config

Code using SerialMenager.Shared;

using System.Windows;

namespace TestLicense

```
{
    public partial class MainWindow : BaseLicense
    {
        public MainWindow()
        {
            InitializeComponent();

            if(this.manager.Check()==true)
            {
                MessageBox.Show("Active");
            }
            else
            {
                MessageBox.Show("need activation");
                this.Close();
            }
        }
    }
}
```

```
}Xaml
Code <lic:BaseLicense x:Class="TestLicense.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:TestLicense"
    xmlns:lic="clr-namespace:SerialMenager.Shared;assembly=SerialMenager"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <Grid>
    </Grid>
```

```
</lic:BaseLicense>Example of usage with custom config
```

Code using SerialMenager.Connection.ServerConf;

using SerialMenager.Connection.ServerConf.MsSQL;

using SerialMenager.Hardware;

using SerialMenager.LicenseModel;

using SerialMenager.LicenseModel.Columns;

using SerialMenager.Manager;

using SerialMenager.RegistryConfig;

```

using SerialMenager.Shared;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Windows;
namespace TestLicense
{
    public partial class MainWindow : Window
    {
        public IManager manager;
        public MainWindow()
        {
            InitializeComponent();
            LicenseQuery query= new LicenseQuery()
            {
                QueryBlockLicense = $"UPDATE SerialManager({LicenseColumns.IsLicenseActive})
VALUES(@blockparam) WHERE {LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} =
@hdd AND {LicenseColumns.SID}=@sid",
                ParametersBlockLicense = new List<SqlParameter>() { new
SqlParameter("@blockparam", false), new SqlParameter("@cpu", HardwareInfo.GetCPUInfo()), new
SqlParameter("@hdd", HardwareInfo.GetHDDInfo()), new SqlParameter("@sid",
LicenseRegistry.Shared().GetSID()) },

                QueryCheckHardware = $"SELECT {LicenseColumns.PID},{LicenseColumns.HID} FROM
SerialManager WHERE {LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd ",
                ParametersCheckHardware = new List<SqlParameter>() { new SqlParameter("@cpu",
HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd", HardwareInfo.GetHDDInfo()) },

                QueryCheckHardwareStatus = $"SELECT {LicenseColumns.IsHardwareActive} FROM
SerialManager WHERE {LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd",
                ParametersCheckHardwareStatus = new List<SqlParameter>() { new
SqlParameter("@cpu", HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd",
HardwareInfo.GetHDDInfo()) },

                QueryCreateLicense = $"",
                ParametersCreateLicense = null,

                QueryDeleteLicense = $"",
                ParametersDeleteLicense = null,

                QueryGetExpiryDate = $"SELECT {LicenseColumns.ExpiryDate} FROM SerialManager
WHERE {LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd AND
{LicenseColumns.SID}=@sid",
                ParametersGetExpiryDate = new List<SqlParameter>() { new SqlParameter("@cpu",
HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd", HardwareInfo.GetHDDInfo()), new
SqlParameter("@sid", LicenseRegistry.Shared().GetSID()) },

                QueryGetLicenseData = $"SELECT
{LicenseColumns.SID},{LicenseColumns.PID},{LicenseColumns.HID},{LicenseColumns.IsLicenseAc
tive},{LicenseColumns.IsHardwareActive},{LicenseColumns.ExpiryDate} FROM License WHERE
{LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd AND
{LicenseColumns.SID}=@sid",
                ParametersGetLicenseData = new List<SqlParameter>() { new SqlParameter("@cpu",
HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd", HardwareInfo.GetHDDInfo()), new

```

```
SqlParameter("@sid", LicenseRegistry.Shared().GetSID()) },
```

```
QueryIsActivated = $"SELECT {LicenseColumns.IsLicenseActive} FROM SerialManager  
WHERE {LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd AND  
{LicenseColumns.SID}=@sid",
```

```
ParametersIsActivated = new List<SqlParameter>() { new SqlParameter("@cpu",  
HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd", HardwareInfo.GetHDDInfo()), new  
SqlParameter("@sid", LicenseRegistry.Shared().GetSID()) },
```

```
QueryIsBlocked = $"SELECT  
{LicenseColumns.IsLicenseActive},{LicenseColumns.IsHardwareActive} FROM SerialManager  
WHERE {LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd AND  
{LicenseColumns.SID}=@sid AND ({LicenseColumns.IsHardwareActive}=False OR  
{LicenseColumns.IsLicenseActive}=False)",
```

```
ParametersIsBlocked = new List<SqlParameter>() { new SqlParameter("@cpu",  
HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd", HardwareInfo.GetHDDInfo()), new  
SqlParameter("@sid", LicenseRegistry.Shared().GetSID()) },
```

```
QueryIsExpired = $"SELECT {LicenseColumns.ExpiryDate} FROM SerialManager  
WHERE {LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd AND  
{LicenseColumns.SID}=@sid",
```

```
ParametersIsExpired = new List<SqlParameter>() { new SqlParameter("@cpu",  
HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd", HardwareInfo.GetHDDInfo()), new  
SqlParameter("@sid", LicenseRegistry.Shared().GetSID()) },
```

```
QueryIsValid = $"SELECT {LicenseColumns.SID} FROM SerialManager WHERE  
{LicenseColumns.PID} = @cpu AND {LicenseColumns.HID} = @hdd AND  
{LicenseColumns.SID}=@sid",
```

```
ParametersIsValid = new List<SqlParameter>() { new SqlParameter("@cpu",  
HardwareInfo.GetCPUInfo()), new SqlParameter("@hdd", HardwareInfo.GetHDDInfo()), new  
SqlParameter("@sid", LicenseRegistry.Shared().GetSID()) }
```

```
};
```

```
manager = SerialManager.Shared(ServerType.MsSQL, new ServerData { ServerIP =  
SerialMenager.RegistryConfig.LicenseRegistry.Shared().GetServerIP(), Username =  
SerialMenager.RegistryConfig.LicenseRegistry.Shared().GetUsername(), Password =  
SerialMenager.RegistryConfig.LicenseRegistry.Shared().GetPassword(), DataBase =  
SerialMenager.RegistryConfig.LicenseRegistry.Shared().GetDataBase() }, query, false, true, true);
```

```
if (this.manager.Check()==true)
```

```
{
```

```
    MessageBox.Show("Active");
```

```
}
```

```
else
```

```
{
```

```
    MessageBox.Show("need activation");
```

```
    this.Close();
```

```
}
```

```
}
```

```
}
```

```
}
```

You can configure the manager almost completely at will

Download dll <https://www.mediafire.com/file/nxy66in1y782a5t/SerialMenager.dll/file>
All the best

SQL script
Code USE [yourdb]
GO

/***** Object: Table [dbo].[SerialManager] Script Date: 11/25/2018 10:12:14 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

```
CREATE TABLE [dbo].[SerialManager](
  [ID] [int] IDENTITY(1,1) NOT NULL,
  [SID] [nvarchar](max) NULL,
  [PID] [nvarchar](max) NULL,
  [HID] [nvarchar](max) NULL,
  [ExpiryDate] [date] NULL,
  [ActivationDate] [date] NULL,
  [IsLicenseActive] [bit] NULL,
  [IsHardwareActive] [bit] NULL,
  CONSTRAINT [PK_SerialManager] PRIMARY KEY CLUSTERED
(
  [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

GO

Check funcions:

Code bool Check(bool debug=false);

bool IsValid();

bool IsActivated();

bool IsBlocked();

bool IsExpired();

bool CheckHardware();

bool CheckHardwareStatus();

DateTime GetExpiryDate();

License GetLicenseData();

LicenseType.Type GetLicenseType(string sid);

void CreateLicense();

void BlockLicense();

void DeleteLicense();

bool CheckSQLConnection();

mandatory information that must be stored in registry is SID,PID,HID,DT and Get,Set
communication with them you can do via LicenseRegistry class

Add this line in app.config it is a secret key for data encryption because I did not set up funcion for
config a secret key over the manager

Code <appSettings>

<add key="SecurityKey" value="05fd924rbclRBQ95" />

</appSettings>