



Free version 1.0.0

## I. О продукте

Telescope – инструмент для удаленного просмотра состояния переменных в разрабатываемой Вами .NET программе (C#, VB.NET, ...). Основная цель нашей программы – дать возможность разработчикам проверять значения интересующих их переменных без остановки исполнения программы, что позволит значительно увеличить гибкость разработки, быстрее находить и исправлять неполадки и сделать разрабатываемые Вами приложения более надежными, удобными для пользования и быстрыми в разработке.

## II. Лицензия

Базовая версия программы Telescope 1.0.0 распространяется без лицензии, что подразумевает право свободно пользоваться программой в личных, образовательных или коммерческих целях.

## III. Установка

Инсталлятор **Telescope.msi** разворачивает программу **Telescope.exe**, собирающую данные с узлов, сборку **TelescopeNode.dll**, (смотри более подробное описание ниже) и текущую документацию. Для удаления программы **Telescope** выполните действие `uninstall` из панели управления: `Control Panel\Programs\Programs and Features`

## IV. Использование

**TelescopeNode.dll** – сборка, отвечающая за получение информации о состоянии объектов в памяти вашей программы.

**Telescope.exe** – программа, собирающая все данные с узлов, работающих в вашей программе и отображающая их в удобном для просмотра виде.

Программа рассчитана на работу с установленным .NET 4.0 и выше.

Для создания привязки к интересующему Вас объекту, следует создать экземпляр **Telescope.TelescopeNode** (узел), передав ссылку на интересующий объект с указанием отображаемого имени узла и имени объекта.

Пример такой привязки в коде может выглядеть так:

```

namespace TestTelescopeConsole
{
    class Program
    {
        static void Main(string[] args)
        {
            Telescope.TelescopeNode n = new Telescope.TelescopeNode(new TestObject(), "NoName", "NoObjectName");

            Console.ReadLine();
        }
    }

    public class TestObject...
    public class InnerObj...
    public struct Str...
}

```

В случае привязки к объектам в WPF-приложении, доступ к которым возможен только из потока окна, привязка должна осуществляться с передачей ссылки на объект `System.Windows.Threading.Dispatcher`, используемый в вашем приложении. В этом случае пример кода, осуществляющего привязку, выглядит так:

```

namespace TestWPFApplication
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        Telescope.TelescopeNode node;

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            Dispatcher.Invoke(new Action(() =>
            {
                node = new Telescope.TelescopeNode(this, "ExampleNode", "ExampleObject", Dispatcher);
            })), null);
        }
    }
}

```

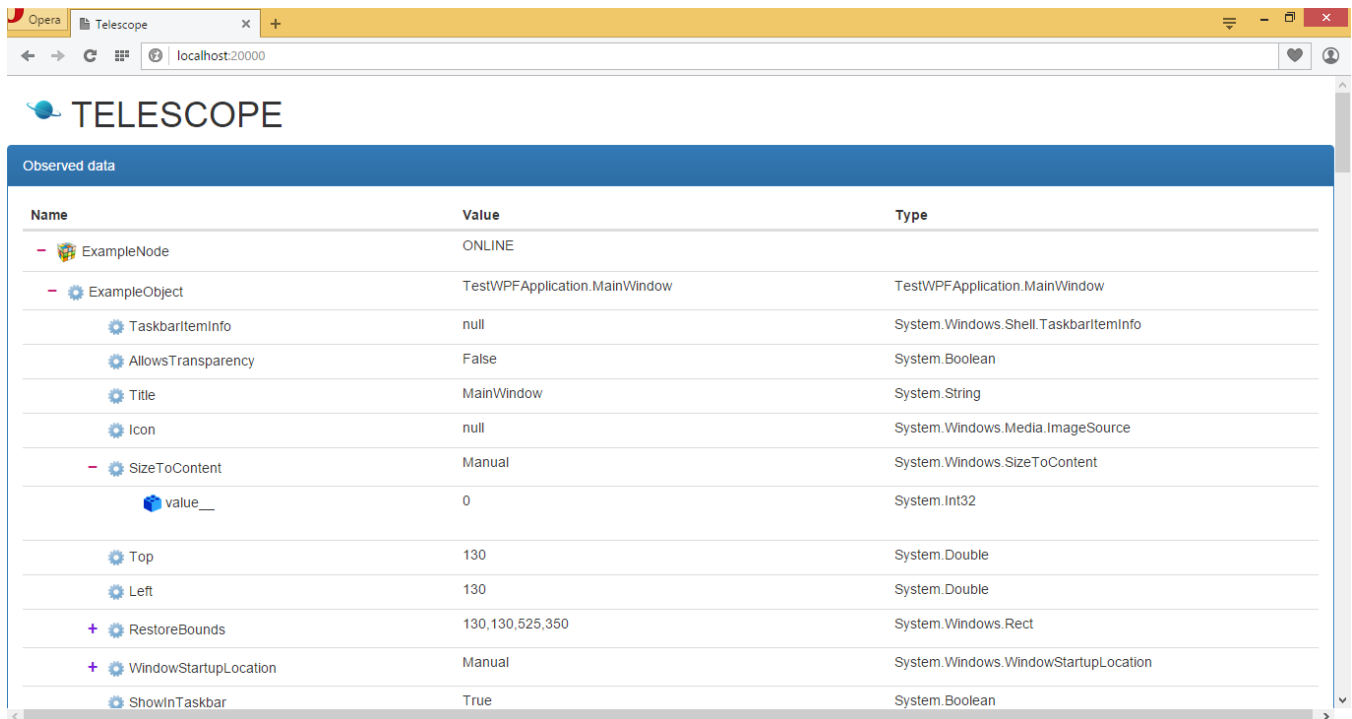
При запуске Вашей программы созданный экземпляр `Telescope.TelescopeNode` будет иметь доступ к получению данных привязанного объекта и может передать эти данные на программу `Telescope`. `Telescope` способен обрабатывать http-запросы, посылаемые на него Вами из браузера.

`Telescope` использует порт, указанный в файле конфигурации `TelescopeConfig.cfg`, который должен находиться в папке `<Telescope folder>\config` (по умолчанию это порт 20000). Экземпляр `Telescope`.

**TelescopeNode** находится в спящем состоянии до тех пор, пока **Telescope** не запущен (**Telescope**. **TelescopeNode** ищет его на порту, указанном в файле конфигурации **TelescopeNodeConfig.cfg** – по умолчанию это порт 20000). Когда соединение между узлом и программой **Telescope** произошло, узлу динамически присваивается порт. Диапазон значений для портов, используемых в узлах, указывается в файле конфигурации программы **Telescope** – **TelescopeConfig.cfg** (по умолчанию используется диапазон 20001-21000).

Если Вы используете браузер на том же компьютере/сервере, на котором запущен Telescope, то http запрос может выглядеть как <http://localhost:20000/>

Пример отображения объектов вашей программы в браузере:



Name	Value	Type
- ExampleNode	ONLINE	
- ExampleObject	TestWPFApplication.MainWindow	TestWPFApplication.MainWindow
TaskbarItemInfo	null	System.Windows.Shell.TaskbarItemInfo
AllowsTransparency	False	System.Boolean
Title	MainWindow	System.String
Icon	null	System.Windows.Media.ImageSource
- SizeToContent	Manual	System.Windows.SizeToContent
value__	0	System.Int32
Top	130	System.Double
Left	130	System.Double
+ RestoreBounds	130,130,525,350	System.Windows.Rect
+ WindowStartupLocation	Manual	System.Windows.WindowStartupLocation
ShowInTaskbar	True	System.Boolean

The screenshot shows the Telescope web interface in a browser window. The page title is "TELESCOPE". Below the title, there is a section titled "Observed data" which contains a table with three columns: "Name", "Value", and "Type".

Name	Value	Type
- NoName	ONLINE	
- NoObjectName	TestTelescopeConsole.TestObject	TestTelescopeConsole.TestObject
- someIntData	System.Int32[]	System.Int32[]
Length	1000	System.Int32
LongLength	1000	System.Int64
Rank	1	System.Int32
+ SyncRoot	System.Int32[]	System.Int32[]
IsReadOnly	False	System.Boolean
IsFixedSize	True	System.Boolean
IsSynchronized	False	System.Boolean
- collection	System.Int32[]	System.Int32[]
[0]	0	System.Int32
[1]	1	System.Int32
[2]	2	System.Int32

Telescope показывает дерево публичных полей и свойств объекта, к которому осуществляется привязка. **Глубина дерева для базовой версии программы равняется 5.**



- пиктограмма корня отображаемого узла



- пиктограмма свойств объекта или объекта, к которому привязан узел



- пиктограмма поля объекта



- пиктограмма корня коллекции

В случае если связь с узлом утеряна (например, содержащая его программа остановлена), данный узел будет помечен как неработающий (OFFLINE). Пример такого узла приведен на следующем скриншоте.

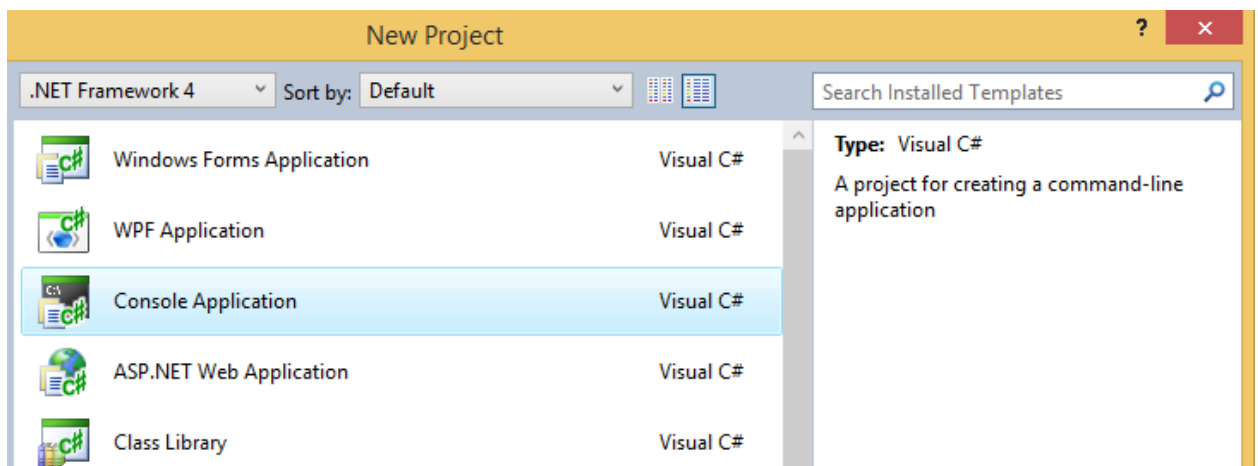
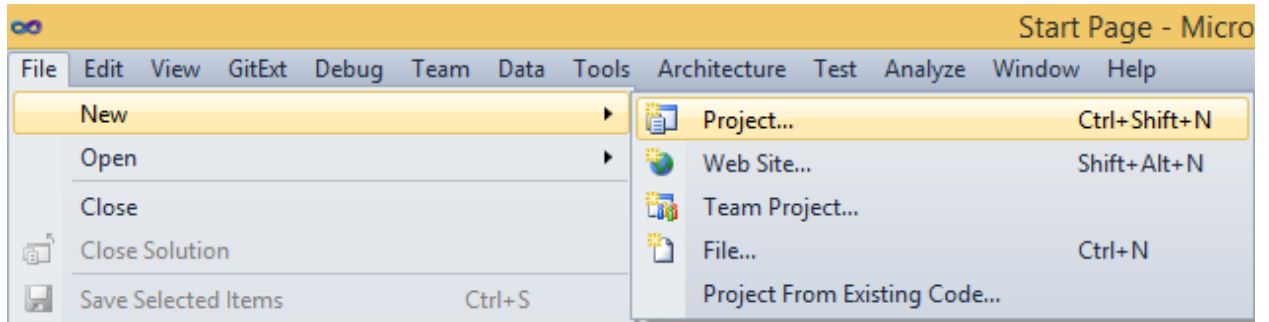
The screenshot shows the Telescope web interface in a browser window. The page title is "TELESCOPE". Below the title, there is a section titled "Observed data" which contains a table with three columns: "Name", "Value", and "Type".

Name	Value	Type
+ ExampleNode	ONLINE	
PORT: 20002	OFFLINE	

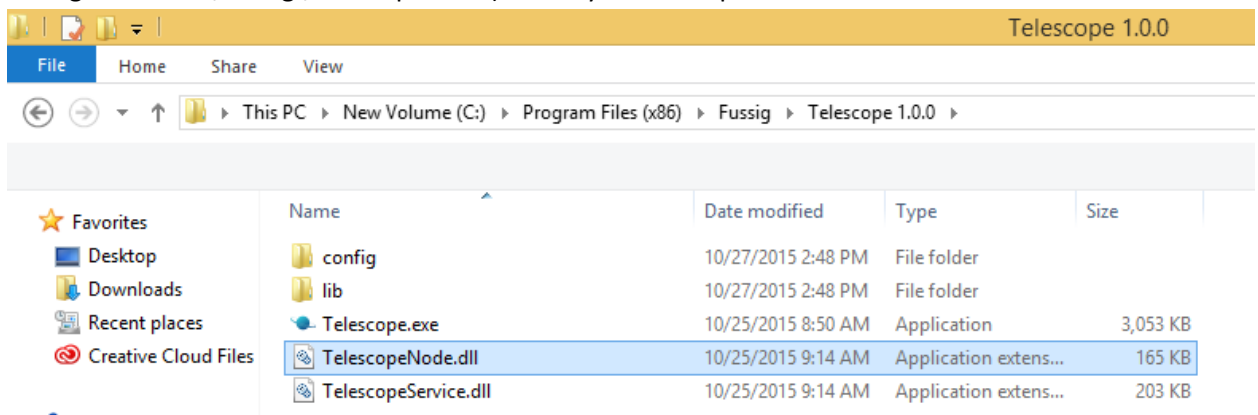
## V. Примеры

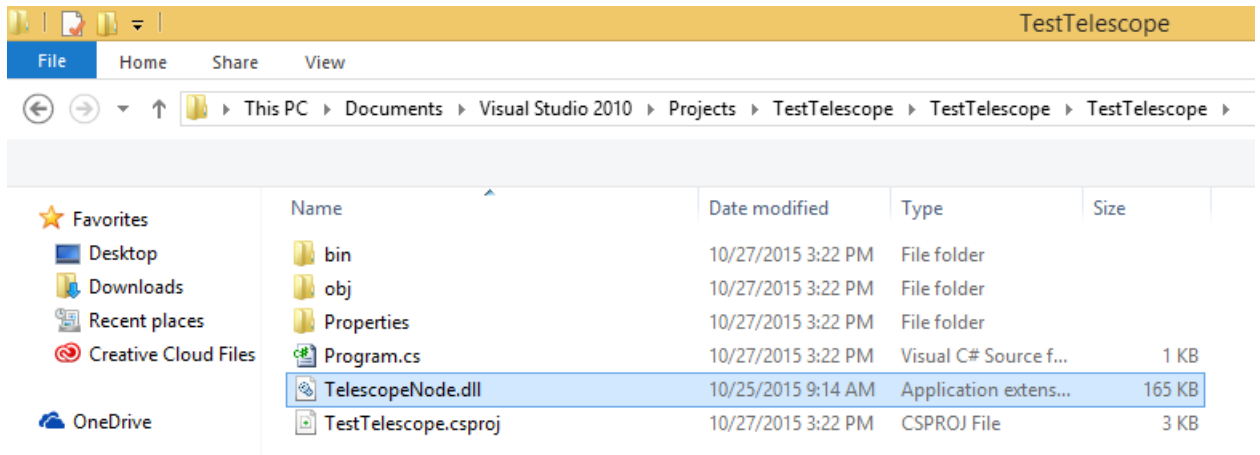
Рассмотрим работу Telescope на примере отображения свойств объектов в простой консольной программе. Ниже представлена пошаговая инструкция по созданию узла TelescopeNode

1. Создайте новый консольный проект в Visual Studio

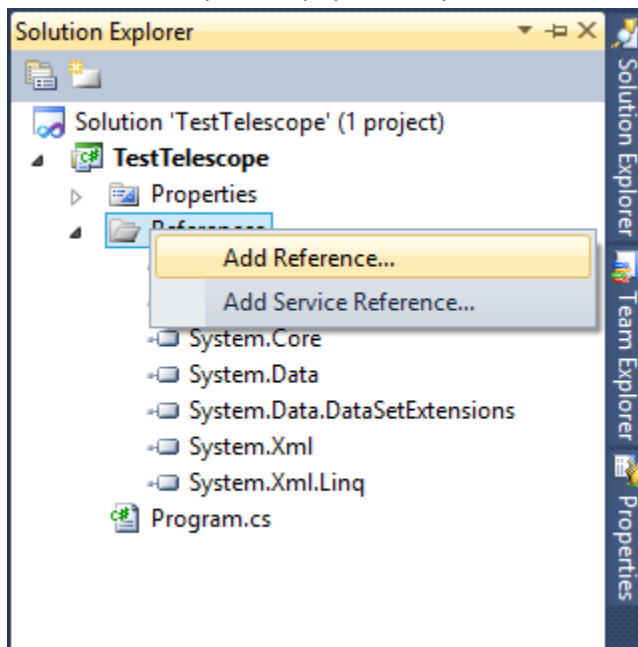


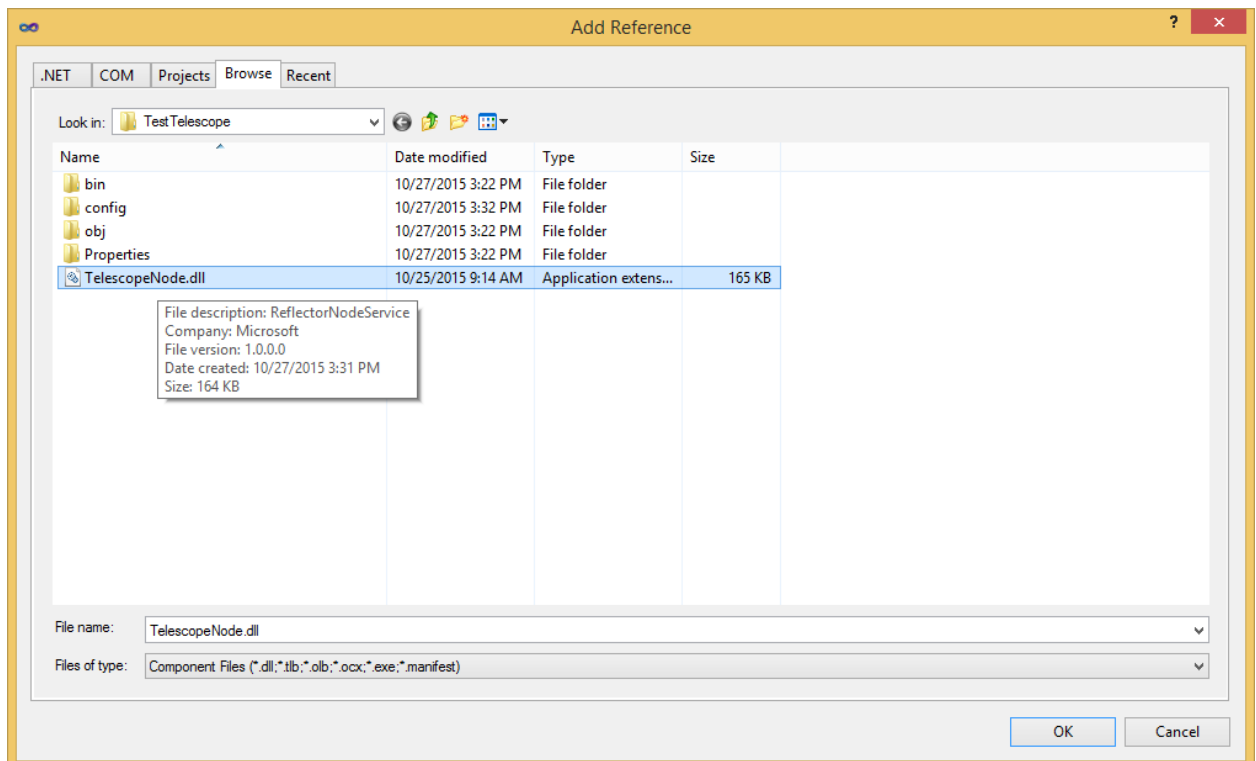
2. Копируйте **TelescopeNode.dll** из папки установки программы (по умолчанию - %ProgramFiles%\Flussig\Telescope 1.0.0) в папку Вашего проекта





3. Создайте ссылку на сборку TelescopeNode.dll в Вашей программе





4. Определите объекты, состояние которых Вас интересует, и создайте экземпляр TelescopeNode, который будет следить за их состоянием

```

namespace TestTelescope
{
    class Program
    {
        static void Main(string[] args)
        {
            MyClass mc = new MyClass();

            Telescope.TelescopeNode node = new Telescope.TelescopeNode(mc, "My node", "mc");

            Console.ReadLine();
        }
    }

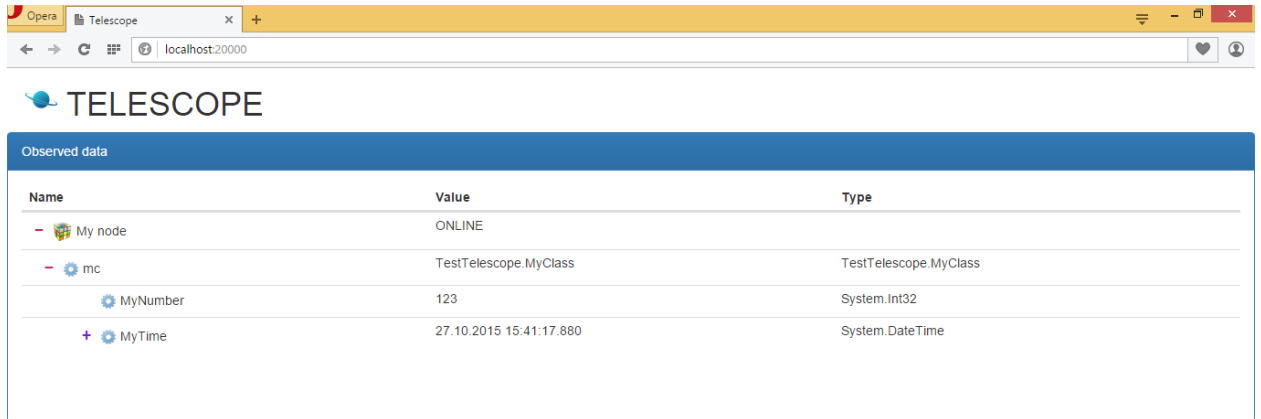
    public class MyClass
    {
        public MyClass()...

        public int MyNumber { get; set; }
        public DateTime MyTime { get { return DateTime.Now; } }
    }
}

```

5. Скомпилируйте проект
6. (Опционально) Создайте конфигурационный файл **TelescopeNodeConfig.cfg** для узла в папке <TargetDir>\config. Файл содержит информацию о порте, используемом программой **Telescope**. Пример конфигурационного файла можно найти в папке config в папке установки программы **Telescope**.

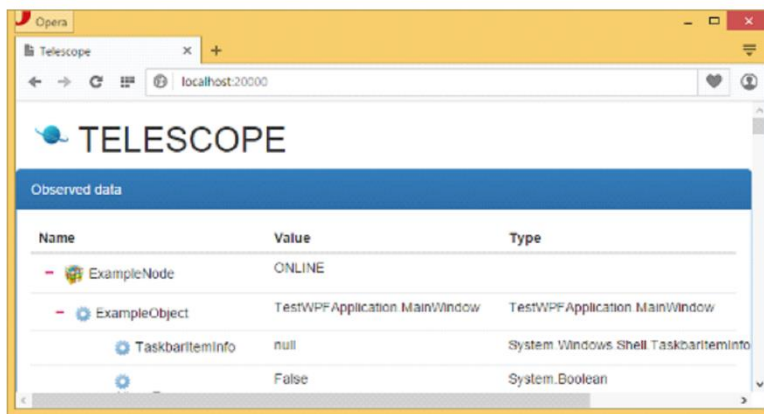
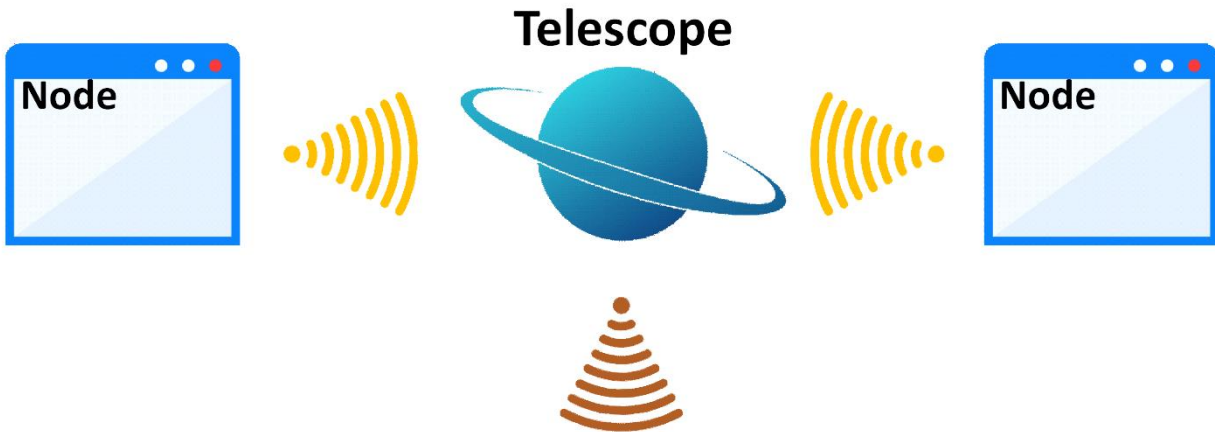
7. Запустите программу **Telescope.exe** (с правами Администратора)
8. Запустите Вашу программу (с правами Администратора)
9. В адресной строке браузера наберите `http://localhost:<номер порта Telescope>`. По умолчанию: <http://localhost:20000/>
10. Вы получили информацию о состоянии объектов в Вашей программе. Например, она может выглядеть так:



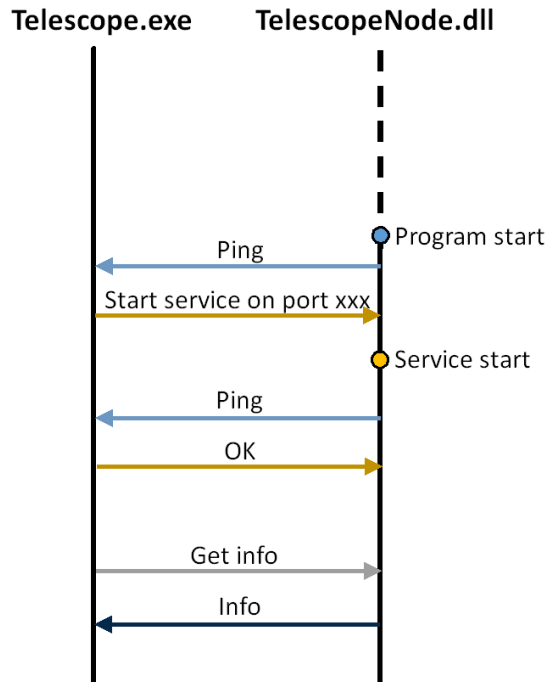
## VI. Схема работы

Telescope служит агрегатором для сообщений от узлов, расположенных в Вашей программе. При запросе от браузера он отправляет запросы на получение информации на узлы, собирает ее и пересылает ее в удобном виде обратно на браузер. Принципиальная схема работы программы **Telescope** представлена а следующем рисунке:





При запуске Вашей программы с внедренным узлом **TelescopeNode.dll** узел неактивен, он не способен передавать сообщения до тех пор, пока не соединится с программой **Telescope**. Когда соединение произошло, Telescope передает на узел номер порта, на котором тот должен работать, после чего узел становится активным и способен передавать информацию об объектах в памяти программы на **Telescope**. Схема взаимодействия между узлом (**TelescopeNode.dll**) и агрегатором (**Telescope.exe**) представлена на следующей диаграмме последовательности сообщений:



## VII. Производительность

Узел **TelescopeNode.dll** получает данные в отдельном потоке и не потребляет ресурсы, используемые в Вашем приложении. Основной объем памяти, потребляемый узлом и основной программой – это пересылаемые сообщения. Мы планируем значительно уменьшить объем потребляемой памяти в будущих версиях программы.