Documentation of the Assistant module "External Communication"

FKE

November 7, 2018

Contents

1 Description

Breanos Connectors is a collection of Nuget packages that were created to provide functionality within the purview of data transmission over different technologies. These include:

- ActiveMqConnector
- BreanosConnectors.Interface
- BreanosConnectors.Kpu.Communication.Common
- BreanosConnectors.Kpu.Communication.Utilities
- BreanosConnectors.SerializationHelper / .Standard
- BreanosConnectors.Utilities
- OpcUaConnector

ActiveMqConnector Encapsulation for an ActiveMq client. It Implemnents the interfaces of BreanosConnectors.Interface to publish its functionality.

BreanosConnectors.Interface Interface project for ActiveMqConnector and possible future connectors.

BreanosConnectors.Kpu.Communication.Common Collection of different classes used for communication with KPUs, e.g. ModelUpdate

BreanosConnectors.Kpu.Communication.Utilities Contains an advanced version of the batchers from BreanosConnectors.Utilities which is used for a collective, structured transmission of ModelUpdates.

BreanosConnectors.SerializationHelper / .Standard Encapsulation of .Net's Xml-Serializer for direct, simple use with strings for deserialization / typeful objects for serialization.

BreanosConnectors.Utilities Contains the batcher classes also found within the Assistant solution.

OpcUaConnector Encapsulation for the OPCFoundation.NetStandard.Opc.Ua opc client.

2 Usage

To use these projects you can either directly use them as dependencies within your own project or alternatively load the nuget packages from the Breanos public Nuget server.

ActiveMqConnector Listing 1 shows a simple case of usage for the ActiveMqConnector, connecting to a server and registering for messages from the queue "myQueue".

```
using System;
using System.Threading.Tasks;
    class Program
{
  static void Main(string[] args)
  {
    BreanosConnectors.Interface.IMqConnector connector = new
    BreanosConnectors.ActiveMqConnector.Connector();
    InitConnector(connector);
    Console.ReadLine();
  }
  public static async Task InitConnector(BreanosConnectors.
    Interface.IMqConnector connector)
  {
    await connector.ConnectAsync("tcp:activemq://127.0.0.1:61616
    ", "admin", "admin");
    connector.Message += OnIncomingActiveMqMessage;
    await connector.ListenAsync("myQueue");

  }

  private static void OnIncomingActiveMqMessage(object sender,
    BreanosConnectors.Interface.OnMessageEventArgs e)
  {
    Console.WriteLine(e.Content);
  }
}
```

Listing 1: ActiveMqConnector example

SerializationHelper Listing 2 shows the code usage of SerializationHelper with Listing 3 showing the output.

```
using System;
using System.Threading.Tasks;
public class DataStorageClass
{
  public int MyInteger { get; set; }
  public string MyString { get; set; }
  public DataStorageClass2 InternalSomething { get; set; }
}
public class DataStorageClass2
```

```
{
  public double SomeDouble { get; set; }
}
class Program
{
  static void Main(string[] args)
  {
    var masterStorage = new DataStorageClass()
    {
      MyInteger = 42,
      MyString = "Hello World",
      InternalSomething = new DataStorageClass2()
      {
        SomeDouble = 3.1415926
      }
    };
    var serialized = BreanosConnectors.SerializationHelper.
    Serialize(masterStorage);
    Console.WriteLine(serialized);
  }
}
```

Listing 2: SerializationHelper code example

```
<?xml version="1.0" encoding="utf-16"?>
<DataStorageClass xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<MyInteger>42</MyInteger>
<MyString>Hello World</MyString>
<InternalSomething>
<SomeDouble>3.1415926</SomeDouble>
</InternalSomething>
</DataStorageClass>
Press any key to continue . . .
```

Listing 3: SerializationHelper output example

3 Installation

See Section 2

4 Configuration

The components are configured and controlled solely through their coding interfaces.

5 Dependencies

Depending on the project, nuget packages may need to be downloaded from the official nuget server.

6 Requirements

ActiveMqConnector For functional use, this requires a running ActiveMq Server.

OpcUaConnector For functional use, this requires a running Opc UA server.

Listings