Table of Contents

All of our C# classes use a standardized style.

Classes

We use the following style for classes:

ClassTemplate.cs

```
/*
 * Copyright 2018 SIB Visions GmbH
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
may not
 * use this file except in compliance with the License. You may obtain
a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the
 * License for the specific language governing permissions and
limitations under
 * the License.
 *
 *
 * History
 *
 * dd.MM.yyyy - [XX] - creation
 */

namespace com.sibvisions.foo
{
    /// <summary>This is the base bar class.</summary>
    /// <author>First Last</author>
    public class Bar
    {
        #region Fields

        /// <summary>the property name for the serializer.</summary>
        public const int TYPE_FOO = 1;

        /// <summary>the value of foo bar.</summary>
        protected Object oValue = null;

        #endregion
```

```
        #region Constructors

        /// <summary>Creates a new instance of Foo</summary>
        /// <param name="pName">the simple name.</param>
        public Foo(String pName)
        {
        }

        #endregion

        #region Properties

        /// <summary>Returns the value object.</summary>
        /// <returns>the value</returns>
        public Object Value
        {
            get
            {
                return oValue;
            }
        }

        #endregion

        #region Interface implementation

        #endregion

        #region Methods

        #endregion

    }    // Foo

    /// <summary>The FooBar handles everything.</summary>
    /// <author>First Last</author>
    sealed class FooBar
    {
        #region Fields

        #endregion

        #region Constructor

        #endregion

    }    // FooBar

}
```

The following rules are defined by this template:

- Variable declaration at the beginning (first constants, then variables)
- Then constructors and initialization methods
- Then the definition of properties
- Then the implementation of interface methods
- Then all overwritten methods (marked with @Override)
- Then all methods of the class
- Sub/Inner classes at the end

- Each parameter of a method is marked using the prefix “p”
- A prefix is also used for instance variables, e.g.:

```
String sValue = "bar";
```

- Important changes are documented in the header, including time stamp and author
- Documentation for the class declaration, ALL methods, properties and instance variables and constants

Interfaces

We use the following style for interfaces:

InterfaceTemplate.cs

```
/*
 * Copyright 2018 SIB Visions GmbH
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
may not
 * use this file except in compliance with the License. You may obtain
a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the
 * License for the specific language governing permissions and
limitations under
 * the License.
 *
 *
 * History
 *
 * dd.MM.yyyy - [XX] - creation
 */
```

```
namespace com.sibvisions.foo
{
    /// <summary>This is the base bar interface</summary>
    /// <author>First Last</author>
    public interface IBar
    {
        #region Method definitions

        /// <summary>Gets the value.</summary>
        /// <returns>the value.</returns>
        public Object getValue();

        #endregion

    }    // IBar
}
```

The following rules are defined by this template:

- Interface methods are defined at the beginning
- Sub/Inner interfaces at the end

- Each interface starts with “I”
- Important changes are documented in the header, including time stamp and author
- Documentation for the interface declaration and ALL methods

Unit Tests

The use of unit tests ensures that basic functionality works as expected. A unit test can never test the entire functionality in all conceivable configurations, but without it the required quality standards cannot be met. We therefore require a working set of unit tests.

Unit test are saved separately from the core source code:

```
<jvxnet>/trunk/net/JVxWin/src/com/sibvisions/foo
<jvxnet>/trunk/net/JVxWin/test/com/sibvisions/foo
```

NUnit is used as the testing framework.

We use the following style for unit tests:

TestTemplate.cs

```
/*
 * Copyright 2010 SIB Visions GmbH
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
may not
```

```
 * use this file except in compliance with the License. You may obtain
a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the
 * License for the specific language governing permissions and
limitations under
 * the License.
 *
 *
 * History
 *
 * dd.MM.yyyy - [XX] - creation
 */

namespace com.sibvisions.foo
{
    /// <summary>Tests the functionality of ...</summary>
    /// <author>First Last</author>
    [TestFixture]
    public class TestBar
    {
        #region Fields

        #endregion

        #region Initialization

        ///<summary>Initializes the unit test.</summary>
        [SetUp]
        public void SetUp()
        {
        }

        ///<summary>Sets values before each test.</summary>
        [TestFixtureSetUp]
        public virtual void FixtureSetUp()
        {
        }

        #endregion

        #region Test Methods

        /// <summary>Tests the ... method.</summary>
```

```
        [Test]
        public void testGet()
        {
        }

        #endregion

    }    // TestBar
}
```

The following rules are defined by this template:

- Variables are declared at the beginning (first constants, then variables)
- Then methods for the test initialization
- Then all test methods (marked with [Test])

- Each test class starts with “Test”
- Each test method starts with “test”
- Each test method begins with
- Documentation for the class declaration, ALL methods, properties and instance variables and constants

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/jvx/join/style_csharp

Last update: **2018/02/06 09:34**