WS-Biometric Devices  
Visual Basic.NET Reference Implementation

April 2013 Update

The April 2013 update is primarily a documentation update as well as a minor “cleanup” of the source code.

# Notice & Disclaimer

The research software provided on this web site (“software”) is provided by NIST as a public service. You may use, copy and distribute copies of the software in any medium, provided that you keep intact this entire notice. You may improve, modify and create derivative works of the software of any portion of the software, and you may copy and distribute such modifications or works. Modified works should carry a notice stating that you changed the software and should note the date and nature of any such change. Please explicitly acknowledge the National Institute of Standards and Technology as the source of the software.

The software is expressly provided “AS IS.” NIST MAKES **NO WARRANTY OF ANY KIND**, EXPRESS, IMPLIED, IN FACT OR ARISING BY OPERATION OF LAW, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT AND DATA ACCURACY. NIST NEITHER REPRESENTS NOR WARRANTS THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT ANY DEFECTS WILL BE CORRECTED. NIST DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF THE SOFTWARE OR THE RESULTS THEREOF, INCLUDING BUT NOT LIMITED TO THE CORRECTNESS, ACCURACY, RELIABILITY, OR USEFULNESS OF THE SOFTWARE.

You are solely responsible for determining the appropriateness of using and distributing the software and you assume all risks associated with its use, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and the unavailability or interruption of operation. This software is not intended to be used in any situation where a failure could cause risk of injury or damage to property. The software was developed by NIST employees. NIST employee contributions are not subject to copyright protection within the United States.

**Specific hardware and software products identified in this open source project were used in order to perform technology transfer and collaboration. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products and equipment identified are necessarily the best available for the purpose.**

# About this Document

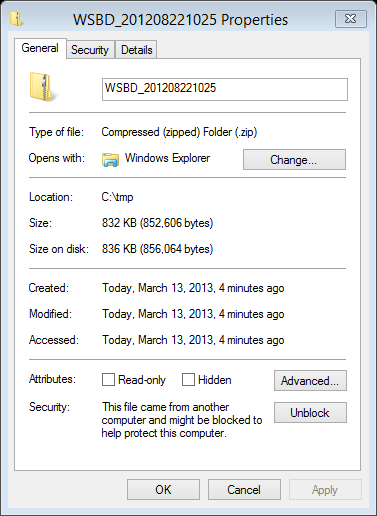
This document is the README file that accompanies the .NET reference implementation of NIST Special Publication 500-288, *Specification for WS-Biometric Devices*. This specification can be downloaded from the NIST Biometric Web Services project website, <http://bws.nist.gov>.

# Audience

This document is intended for software developers interested in the implementation of a web service that conforms to the WS-Biometric Devices specification. It assumes that the reader is familiar with the Microsoft .NET platform as well as the WS-Biometric Devices protocol.

# ZIP File

The reference implementation is a ZIP file contains this file, a Microsoft Visual Studio 2010 solution, and seven Visual Basic .NET projects. If you have downloaded the reference implementation from a source you trust, unblock the ZIP file. To do so

1. In the Windows File Explorer, navigate to the location of the downloaded file
2. Right click on the downloaded file. From the shortcut menu, click “Properties”
3. At the bottom of the “General” tab you will see a section labeled “Security.” There may be a statement such as “This file came from another computer and might be blocked to help protect this computer.” If so, click the Unblock button, and then click the OK button to dismiss the properties window. Unblocking the file will make it easier for Visual Studio to consume the archive’s contents.

# Quick Start

Use the following simplified procedure to implement your own WS-Biometric Devices service.

1. Open the solution file.
2. Create a new project.
3. Add a reference to the *SensorService* and *ISensorService* projects to the new project.
4. Create a new class that inherits from *SensorService*.
5. Implement missing methods.
6. Deploy!

# Contents Summary & Architectural Overview

This package contains a Visual Studio.NET Solution with seven Visual Basic.NET projects. The following four projects are core to the reference implementation.

1. **Infrastructure.** This project contains classes for threading and synchronous and asynchronous job management used in the reference implementation. This reference implementation used neither the .NET thread pool nor the Task Parallel Library—this is so that threads could be forcefully interrupted, instead of relying on cancellation tokens or other cooperative multithreading techniques. This is a valuable feature for cancelling operations that are tied to hardware; cooperative cancellation cannot always interrupt a thread blocked due to a hardware operation.
2. **ISensorService.** This project contains the data type and .NET interface to a WS-BD web service. This interface (and its accompanying data types) is exposed as a web service through the Windows Communications Foundation WCF REST bindings (specifically, the WCF *DataContractSerializer* and *WebInvoke* attributes). The classes in this project map directly to the WS-Biometric Devices specification’s Data Dictionary (§3).
3. **SensorService.** This project contains an abstract implementation that can be used to help implement a WS-Biometric Devices service. Implementations for the registration, locking, and cancellation operations are implemented here in addition to delegating other operations to use the threading classes from Infrastructure. A storage provider interface and a basic implementation to use the local file system also are in this project.
4. **BaselineSensorService.** This project contains a functional WS-BD service to facilitate in testing the integrity and functionality of the solution’s inner workings and infrastructure. It is not intended to be used as a base class for real sensor services since it contains methods to force various failures modes.

Figure 1 is a visual depiction of the project dependencies.

Figure . Core project dependencies.

Each rounded rectangle represents a project and library (DLL). Each layer in the diagram depends on the layer directly below it; specifically, the *BaselineSensorService* project depends on the *SensorService* project. The *SensorService* project depends on both the *ISensorService* and *Infrastructure* projects. There are no dependencies between the *ISensorService* and the *Infrastucture* projects.

The remaining three projects are MSTest-based unit and integration tests. These tests service serve to (a) demonstrate how WS-BD might be used in practice and (b) verify the correctness of the reference implementation.

1. **InfrastructureTests**
2. **SensorServiceTests**
3. **BaselineSensorServiceTests**

Each test library (unsurprisingly) contains unit and/or integration tests for its corresponding core library. The *ISensorService* library is primarily an interface definition, so it does not have a separate test library.

# Common Problems Running the Test Suite

The following information may be helpful if you are having difficulties running the tests in the test projects.

* The *SensorServiceTests* and *BaselineSensorServiceTests* include integration tests that require the hosting of a WS-Biometric Devices service. Microsoft Windows operating systems that are Vista and later operating systems require that permissions be set appropriately to allow non-administrators to host web services. This can be accomplished with the “netsh http” command. Specifically, use the command

netsh http add urlacl url=*[URL]* user=*[user] ]*

where *[URL]* and *[user]* are replaced with the desired endpoint URL and user accordingly.

* Many of the tests will deliberately throw exceptions. If you have Visual Studio to automatically trigger a breakpoint on one of these planned exceptions, your debugger may be triggered when such an exception is thrown.
* Some of the threading tests (InfrastructureTests/ThreadJob) are time sensitive. Consequently, the results of these tests may be sensitive to the environment within which they are run; particularly if the environment is already under load. Although we have attempted to run these tests within a variety of configurations, the thresholds that determine pass/fail criterion *may* need to be adjusted to your particular environment.

# Contact

We would like to hear how others are using WS-Biometric Devices and/or the NIST reference implementation. Please submit all questions, comments, suggestions, or issues to [wsbd@nist.gov](mailto:wsbd@nist.gov).