

NIST/ITL's Biometric Application Programming  
Interface (BioAPI) Conformance Test Suite (CTS)  
Implementation (the BioAPI Test Environment)

Beta Implementation V1.1  
February 28, 2006

## User Guide

National Institute of Standards and Technology (NIST)  
Information Technology Laboratory (ITL)  
Computer Security Division (CSD)

Fernando Podio  
NIST/ITL CSD

Mark Jerde (NIST Guest Researcher)  
The Biometric Foundation

## Table of Contents

Introduction.....	3
Requirements .....	3
How to Find the Java Version Installed.....	4
Downloading and Installing Java.....	6
BTE Installation.....	7
Running The BTE.....	8
GUI Description.....	10
Main Window Elements .....	10
Main Dialogs.....	19
Special Features .....	24
Controlling Displayed BSPs .....	24
Controlling The Assertions Listed.....	24
_LastLog.xml .....	25
Message Boxes.....	25
Supported Functions .....	25
Software Distribution - File List .....	28

# NIST/ITL's Biometric Application Programming Interface (BioAPI) Conformance Test Suite (CTS) Implementation (the BioAPI Test Environment)

## Beta Implementation V1.0 - January 4, 2006

### User Guide

#### Introduction

A companion document, NIST/ITL's Biometric Application Programming Interface (BioAPI) Conformance Test Suite (CTS) Implementation (the BioAPI Test Environment) Beta Implementation V1.0 - January 4, 2006 – "Overview" provides a background on the NIST Biometrics Standards program. It refers to American National Standard InterNational Committee for Information Technology Standards (ANSI INCITS) 358-2002, the BioAPI specification and the BioAPI conformance testing methodology standard under development in INCITS Technical Committee M1 – Biometrics. It discusses the need for conformity assessment efforts in support of the BioAPI standard and other biometric interoperability and data interchange standards. The document provides an overview of NIST/ITL's BioAPI Conformance Test Suite, its development history, its overall architecture and a description of its components. A brief reference to the organizations involved in the development and testing of NIST/ITL BioAPI CTS is also included.

This user guide provides computer requirements for installing the BioAPI Test Environment (BTE). It contains information on how to download and install Java, it describes the BTE installation procedure, and how to run the BTE. An informative brief walkthrough is provided. A list of currently supported BioAPI and BioSPI functions and a software distribution - file list is also provided.

#### Requirements

The following are the computer requirements for installing the BTE.

1. *Microsoft Windows 2000 Professional* or higher. While the BTE may run on lower versions of *Windows* the lower versions have not been tested and they are not supported.
2. The Java Runtime Environment 1.4.2 or higher.

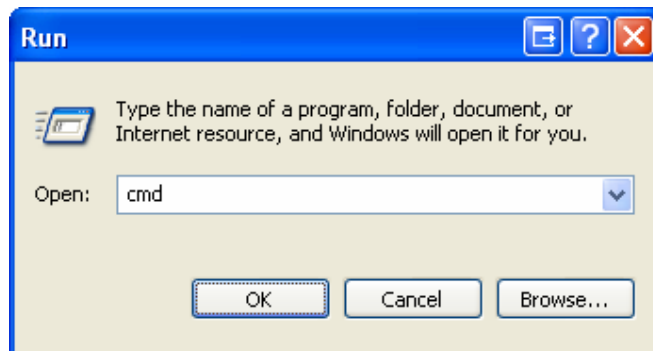
## **How to Find the Java Version Installed**

To determine the version of Java installed on a computer follow these steps for both *Windows 2000 Professional* and *Windows XP Professional*.

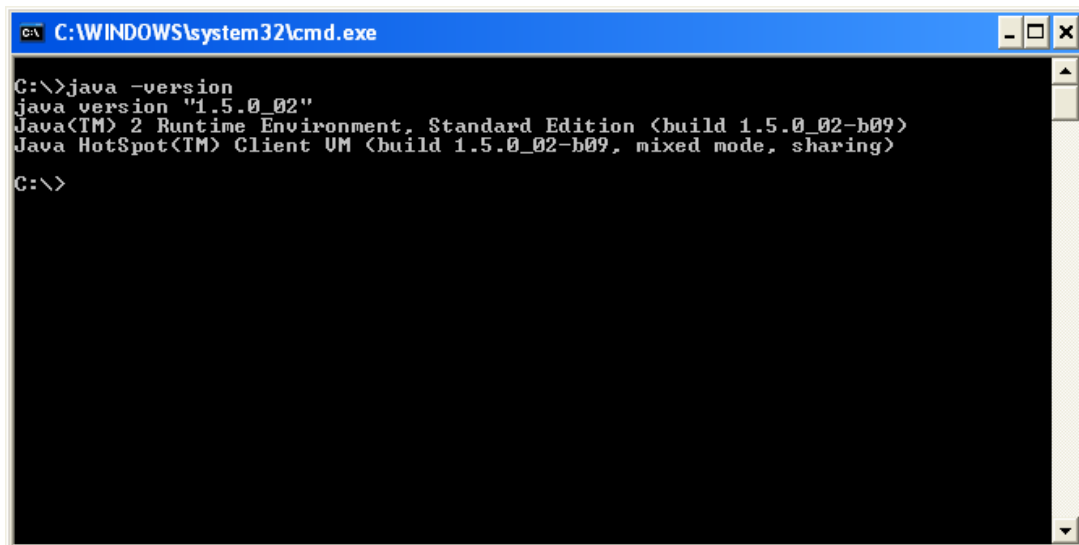
1. From the *Start* button select *Run...*



2. In the *Run* dialog box type "cmd" (without the quotes) in the *Open* textbox and click the *OK* button.



3. In the Command window, enter the command "java -version" (without the quotes) and press the *Enter* key. The following is a screenshot of a Java installation that exceeds the minimum Java requirements. If your computer does not have Java installed, or the Java installation is less than the minimum requirements, **please install an acceptable version of Java.**



```
C:\WINDOWS\system32\cmd.exe
C:\>java -version
java version "1.5.0_02"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_02-b09)
Java HotSpot(TM) Client VM (build 1.5.0_02-b09, mixed mode, sharing)
C:\>
```

### **Downloading and Installing Java**

At the time of this writing the latest version of Java is at the URL  
<http://java.sun.com/j2se/1.5.0/download.jsp>

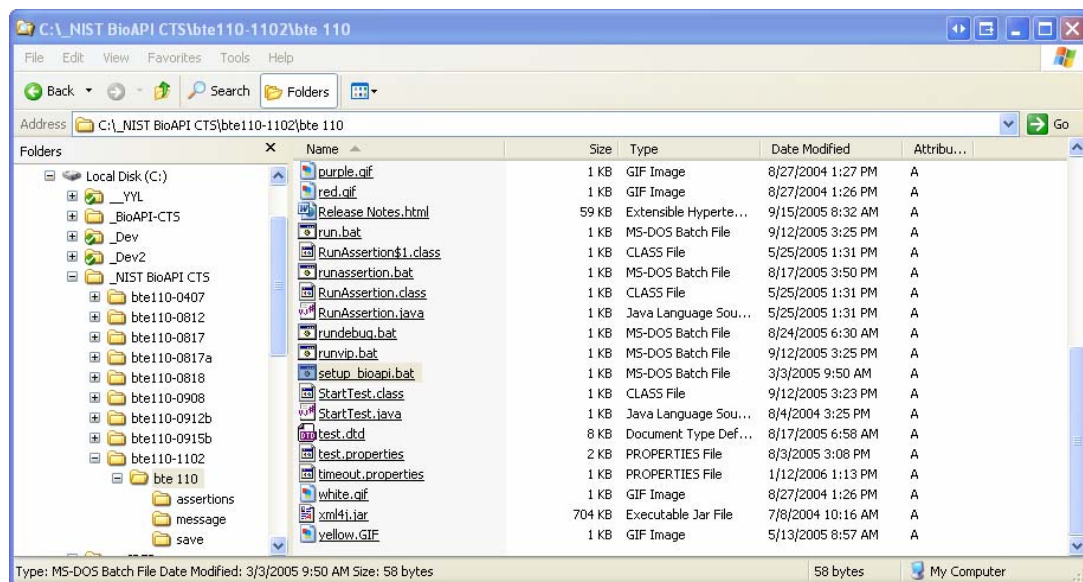
Follow the instructions on the web page to install Java.

To get the latest version of Java go to this URL  
<http://java.sun.com/>

and select *J2SE (Core/Desktop)*. Follow the instructions on the web page to install Java.

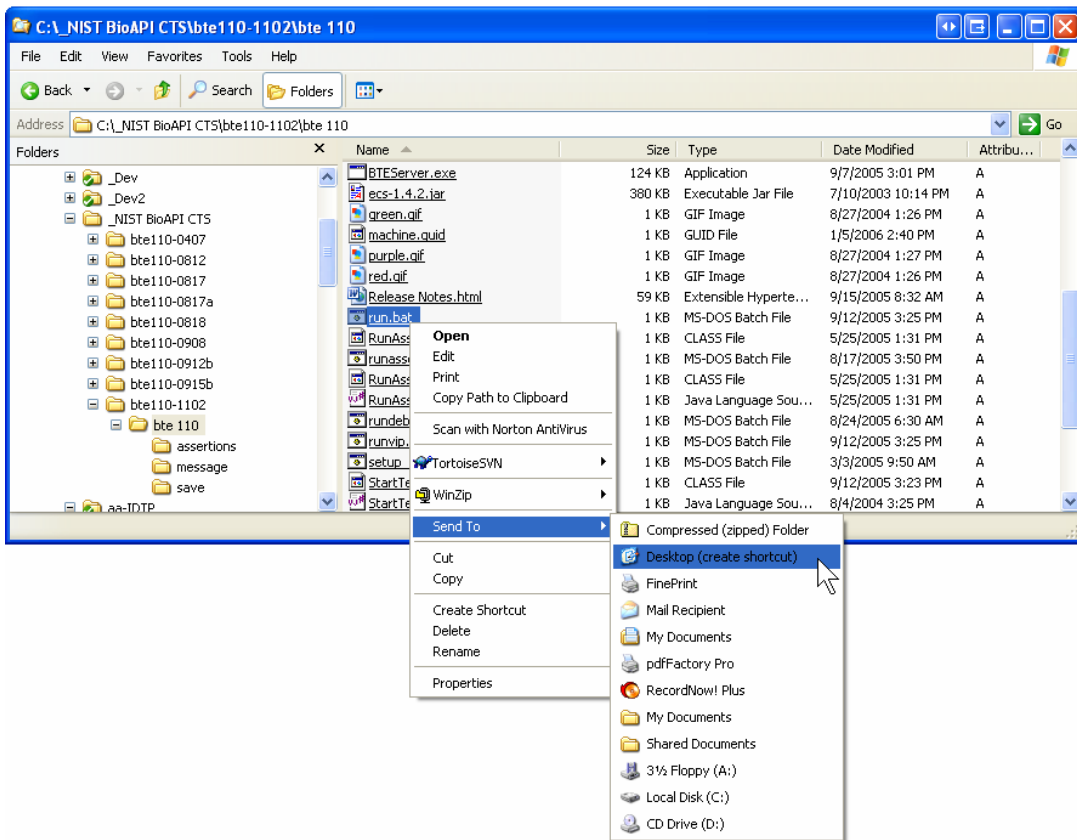
## **BTE Installation**

1. Make sure your computer meets the minimum requirements, including having an acceptable version of Java installed.
2. Extract the BTE.zip file to a folder on your hard drive.
3. In *Windows Explorer*, navigate to the BTE\* folder and double-click **setup\_bioapi.bat**. This command installs the enhanced version of the BioAPI 1.1 framework on your computer.



**Important!** Be sure to run **setup\_bioapi.bat** after every BSP you install. This reinstalls the enhanced framework so the BTE will work properly.

You may want to create a desktop icon for the **run.bat** and/or the **runvip.bat** commands. (These are described in the topic **Running The BTE**.) To create a desktop icon, right-click the file and from *Send To* menu item select *Desktop (create shortcut)*.



## Running The BTE

### **run.bat. and runVIP.bat.**

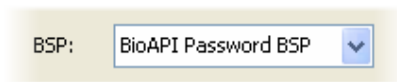
Two different modes for executing BTE are offered. The generic mode (using run.bat) is recommended for non-demonstration, testing purposes. In this mode, the BTE application will be minimized during test execution, and then re-opened when the results are returned. In this mode, focus is provided more elegantly to the user input screens when necessary, and other applications may be open with minimal impact to the user's desktop and interaction with it. The VIP mode (using runVIP.bat) is intended for demonstration mode. It is strongly recommended that all other applications are closed when using this mode. It does not minimize the BTE application so that the user may observe the results display as they are returned. If other applications are open, however, during the execution of the assertions, the BTE GUI will "disappear" behind other applications (sent to the back of the desktop).

The GUI is described beginning with the topic **Command Window**.

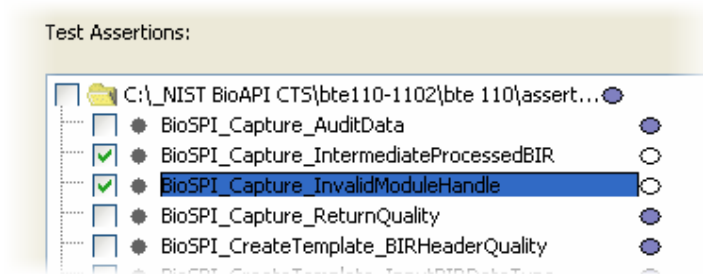


## Brief Walkthrough

1. From the unzipped BTE folder, launch either **run.bat**. OR **runVIP.bat**. The **BTE Main Window** appears.
2. From the *File* menu select **Properties**. The **BTE Properties dialog** appears. At a minimum fill in the four (4) required fields. To show a safety feature of the BTE enter "asdf" or a similar nonsense word in the **Product Name** field. Click the *OK* button when you are done.
3. In the Select BSP dropdown box select **BioAPI Password BSP**.

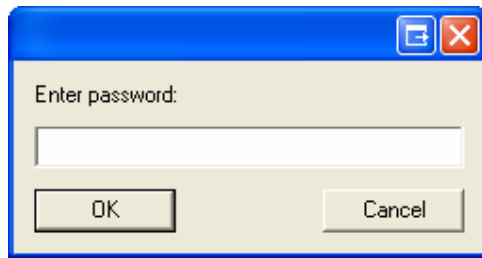


4. Select a couple of assertions in the **Test Assertion Panel**.

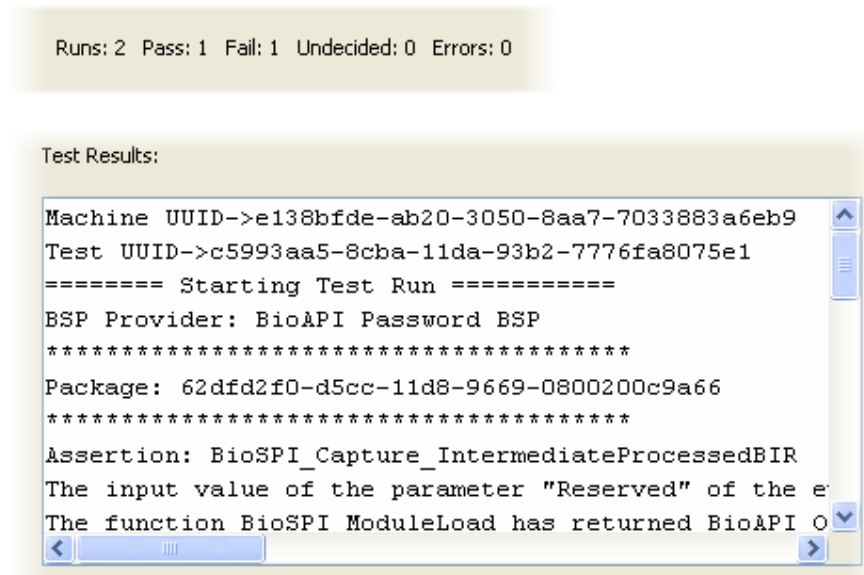


**Note:** Having selected more than one assertion to execute is only a convenience relative to checking the boxes. This will not circumvent the need to present the password or biometric to execute an assertion. The user will be asked to present the biometric for each assertion executed as necessary.

5. Click the **Run Test** button.
6. The **Product Name dialog** will appear, indicating the XML Log may have the incorrect BSP. Click the *Change the Product Name and Continue Testing* button.
7. Depending on the assertions you selected, the BioAPI Password BSP dialog box may appear. If it does show up enter a password and click the *OK* button.



8. When the assertions have completed observe the **Results Counter** and the **Test Results**.



9. You can view and/or save the XML Log file using the **Test Log Menu**.

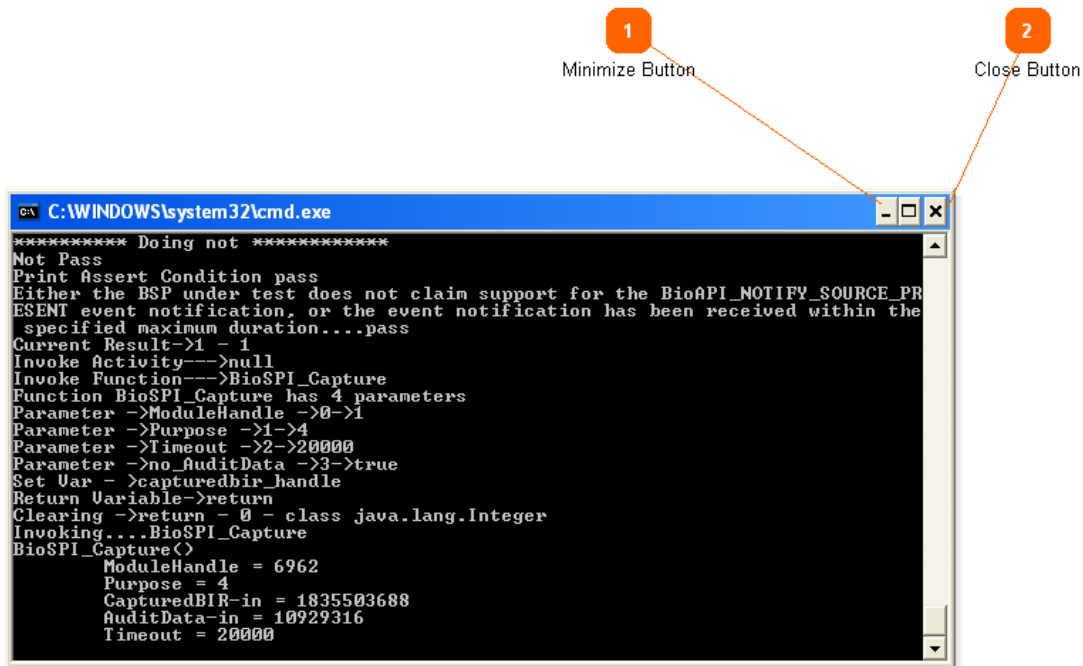
### **GUI Description**

The structure of the BTE GUI, including the **Main Window Elements** and **Main Dialogs** is addressed in this topic.

### **Main Window Elements**

The Main Window Elements will be addressed in this topic. They include: (a) Command Window; (b) Application Main Window; (c) Test Assertions Panel; (d) File Menu; (e) Test Log Menu; and (f) Help Menu

## Command Window



The BTE displays two windows, the Command Window shown here, and a Graphical User Interface (GUI) window described [elsewhere in the manual](#).

The Command Window displays status and messages of the assertions as they run. If you do not want to see this window you can minimize it.



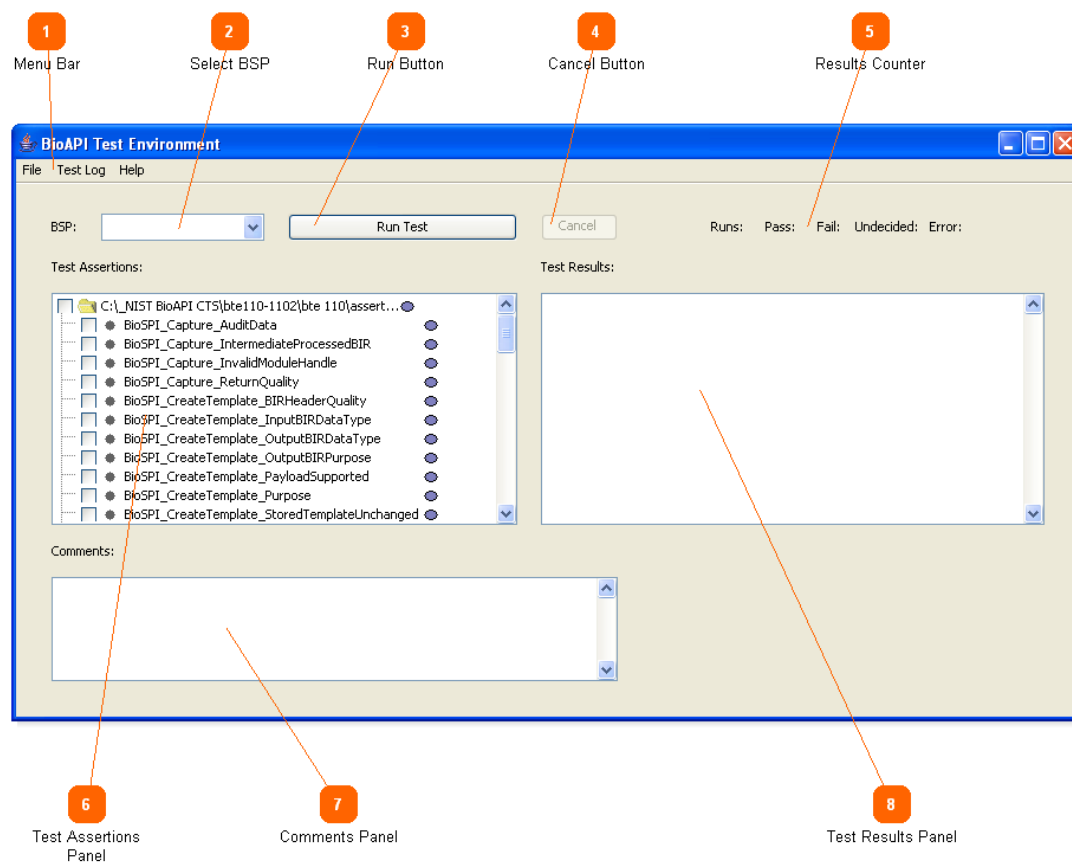
Click this button to minimize the command window.



Click this button to exit the BTE.

You can also exit the BTE by clicking the close button on the **BioAPI Test Environment** window.

## **BTE Main Window**



This is the main window of the BTE. It allows you to select any installed BSP, choose one or more assertions to execute, run the assertions, and optionally save the XML log file.

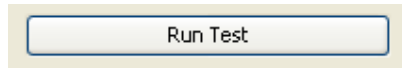


Change BTE settings and save log files.



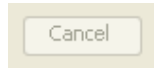
Select a BSP to test. Note: To limit the BSPs that show up in this dropdown box, such as for demonstration purposes, see the topic **Controlling Displayed BSPs**.

**3** **Run Button**



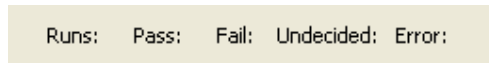
Click this button to run the selected assertions.

**4** **Cancel Button**



When assertions are running this button is enabled and clicking it stops testing.

**5** **Results Counter**



It displays numerical results of the last assertions run.

**Runs:** Total number of assertions run.

**Pass:** The number of assertions that passed.

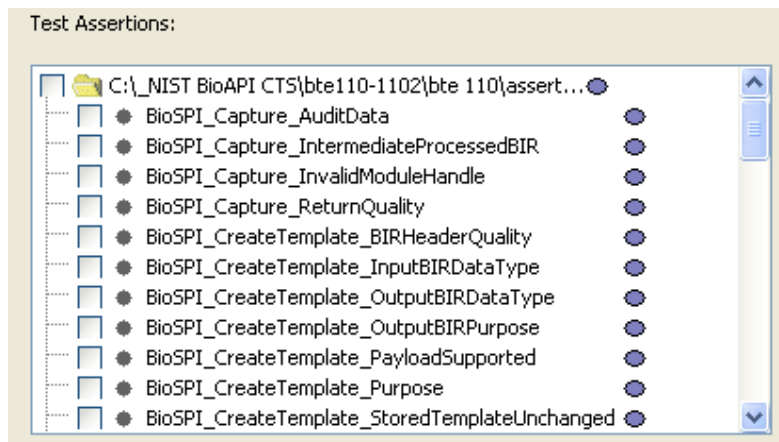
**Fail:** The number of assertions that failed.

**Undecided:** The number of undecided assertions.

**Error:** The number of assertions that caused an error.

This display is reset every time the *Run Test* button is clicked.

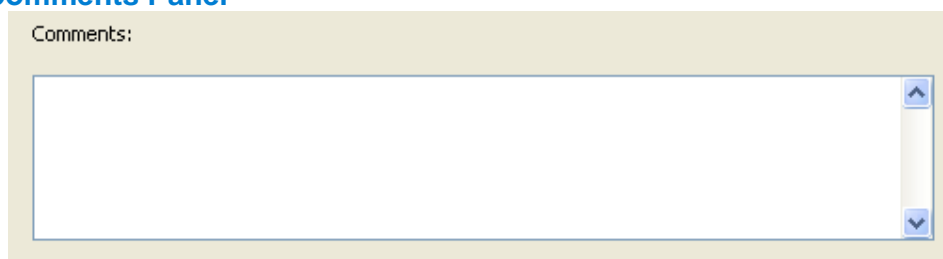
**6** **Test Assertions Panel**



It allows the user to select one or more assertions to run and provides visual feedback on the results of the assertions when run. For more details see the topic [Test Assertions Panel](#).

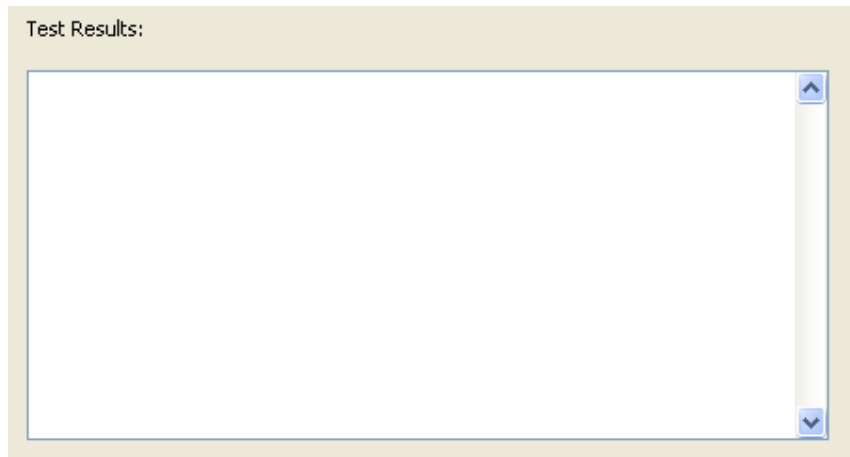
**Note:** Sometimes you may want to see just a few assertions in this panel. See the topic [Controlling The Assertions Listed](#).

## 7 Comments Panel



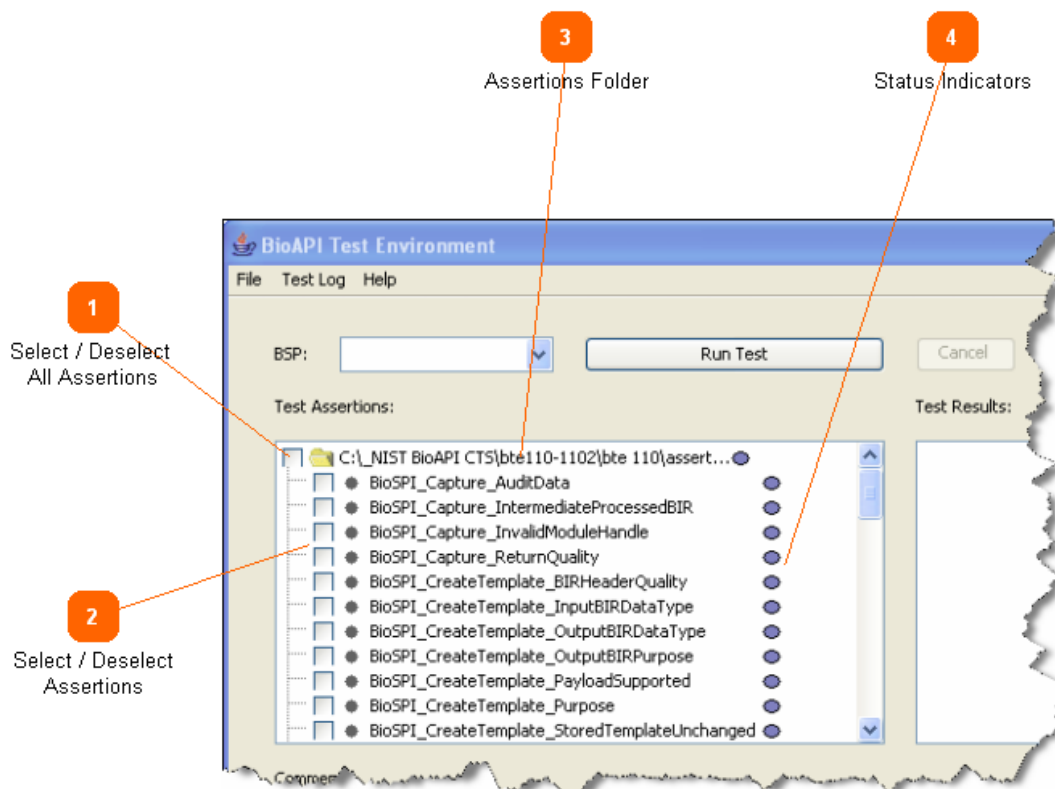
Enter comments to be included in the XML log file.

## 8 Test Results Panel



Displays test results as the assertions are being run. The displayed information is similar to the information in the XML log file.

### **Test Assertions Panel**

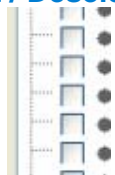


**Note:** Sometimes you may want to see just a few assertions in this panel. See the topic [Controlling The Assertions Listed](#).

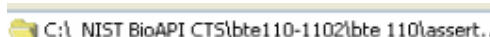
**1** **Select / Deselect All Assertions**

 Click this box to select or deselect all assertions.


**2** **Select / Deselect Assertions**







 Click these boxes to select or deselect individual assertions.

**3** **Assertions Folder**

 Displays the folder where the assertions are installed.

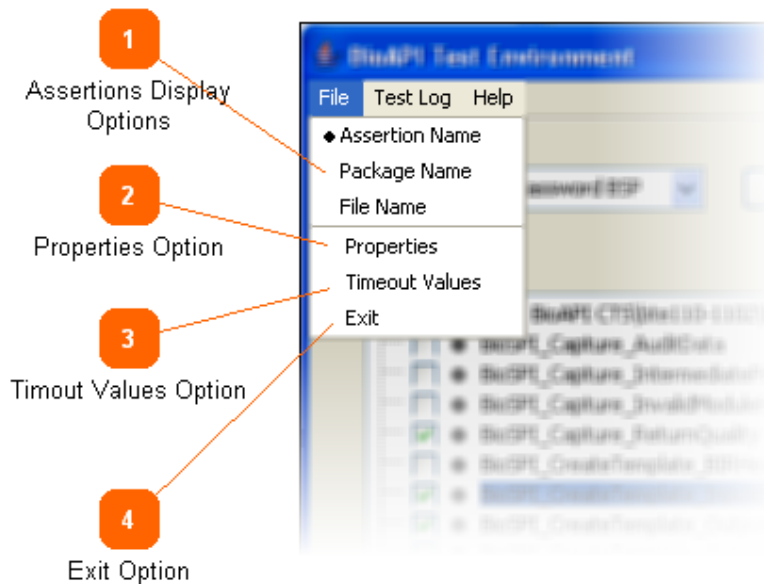
**4** **Status Indicators**

 These ovals change color to show the status of the assertion.

-  **Purple:** The assertion is not selected.
-  **White:** The assertion is selected.
-  **Green:** The assertion passed.
-  **Red:** The assertion failed.
-  **Yellow:** The results of the assertion are undecided.
-  **Black:** The assertion encountered an unrecoverable error while running.



## File Menu



The File menu allows you to change how the assertions are listed in the Test Assertions Panel, change data that will be included in the XML log files, change Timeout Values, and Exit the BTE.

### 1 **Assertions Display Options**

- ◆ Assertion Name
- Package Name
- File Name

Use these menu options to display the assertions in the Test Assertions Panel by Assertion Name, Package Name or File Name. (The Package Name and Assertion Name are contained in the XML assertions themselves.)

**Note:** See the topic **Controlling The Assertions Listed** for an example of why you may want to use the File Name option.

### 2 **Properties Option**

- Properties Select this option to display the **BTE Properties** dialog.

### 3 **Timeout Values Option**

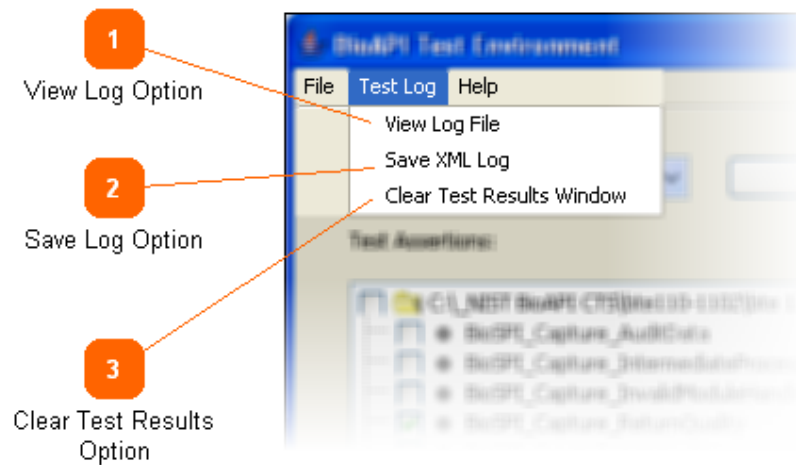
Timeout Values Select this option to display the **BTE Timeout Values** dialog.

4

### Exit Option

Exit Select this option to close the BTE.

### **Test Log Menu**



The Test Log menu lets view and save the XML log, and clear the test results window.

1

### View Log Option

View Log File

Select this option to **view the XML log**.

**Note:** If the BTE crashes the XML log of all the assertions run prior to the crash are available. See the topic **LastLog.xml**.

2

### Save Log Option

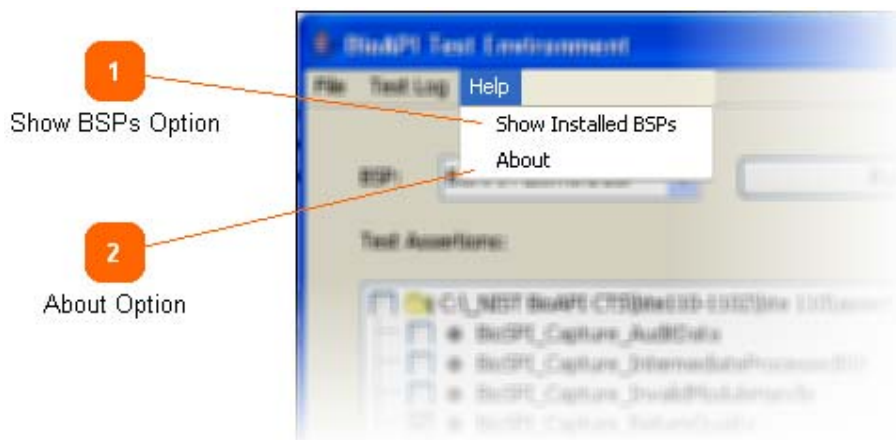
Save XML Log

Select this option to save the XML log to a file.

**3** **Clear Test Results Option**  
Clear Test Results Window

Select this option to clear the Test Results Panel. Note: This option does not affect the XML log in any way.

**Help Menu**



The Help menu lets you see a list of the installed BSPs and display the BTE's About dialog.

**1** **Show BSPs Option**  
Show Installed BSPs Select this option to see a **list of the installed BSPs.**

**2** **About Option**  
About Select this option to display the BTE version information.

**Main Dialogs**

The Main Dialogs windows include: (a) BTE Properties; (b) Product Name; (c) BTE Timeout Values; (d) View XML Log; and (e) Installed BSPs

## **BTE Properties**

The screenshot shows the 'BTE Properties' dialog box with four callouts: 1 points to the 'Name' field in the 'Testing Lab' section; 2 points to the 'Product Name' field in the 'Biometric Product' section; 3 points to the 'CTS ID' field in the 'Biometric Product' section; and 4 points to the 'Tester' field in the 'Testing Information' section. The dialog box includes sections for 'Testing Lab', 'Vendor', 'Biometric Product', and 'Testing Information', each with various input fields. A legend at the bottom left indicates that fields marked with an asterisk (\*) are required.

The *BTE Properties* dialog box is displayed by selecting the menu item **File / Properties** on the main BTE window. All the entries on this dialog box are entered in the XML log file.

After extensive use of early versions of the BTE, it was decided that four (4) elements on this dialog box are essential to ensuring that the XML log file generated by running the assertions has sufficient information to unambiguously identify the logged information. For example, if the log does not contain the specific BSP tested, or the BSP is incorrect, the log is worthless. For information about how the BTE helps to make sure the **Product Name** is correct see the **Product Name** entry.

The four (4) required entries are described below.

1 **Name**  
(\*) Name:

Enter the name of the computer on which the assertions are run. Typical entries include "Lab Computer 'Spitfire'" and "Dell 8300 Laptop 'Tangent.'"

**2** **Product Name**  
(\*) Product Name:

Enter the name of the BSP being tested. Typical entries include "BioAPI Password BSP" and "BioAPI Dummy BSP"

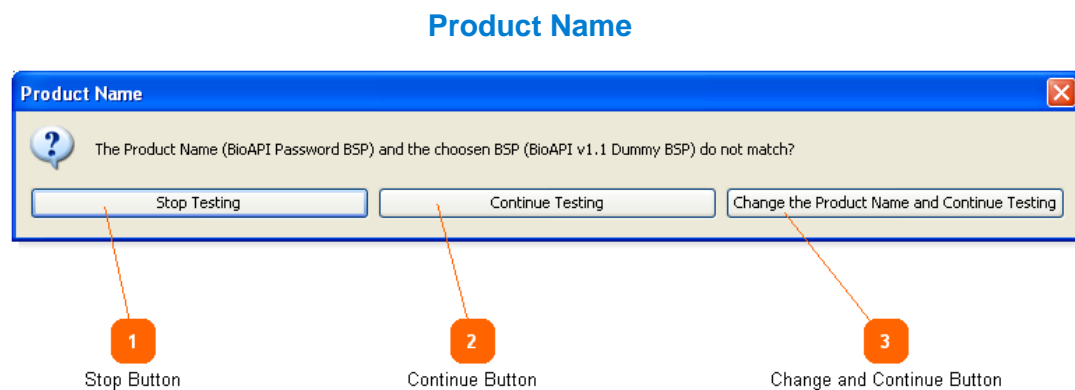
**3** **CTS ID**  
(\*) CTS ID:

Enter the version of the BTE used to perform the testing.

**4** **Tester**  
(\*) Tester:

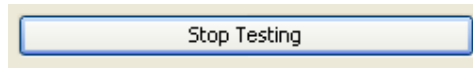
Enter the name of the person running the tests.

### **Product Name**



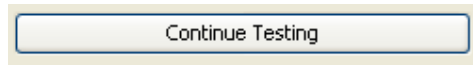
When you click the *Run Test* button the BTE displays this warning dialog if the Product Name on the *BTE Properties* dialog box does not match the selected BSP. It does this to help make sure the BSP in the XML log file is correct.

**1** **Stop Button**



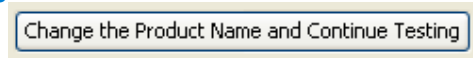
Click this button to stop testing.

**2** Continue Button



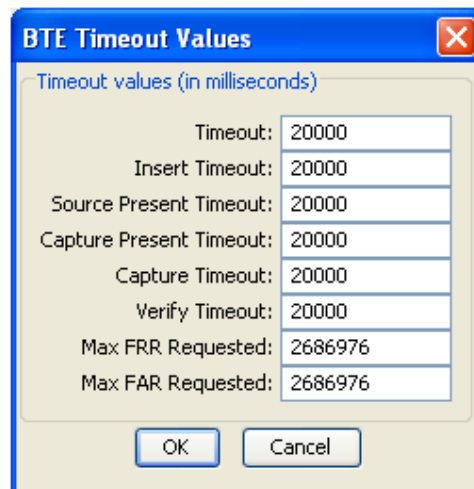
Click this button to continue testing, even though the Product Name is different than the Selected BSP Name.

**3** Change and Continue Button



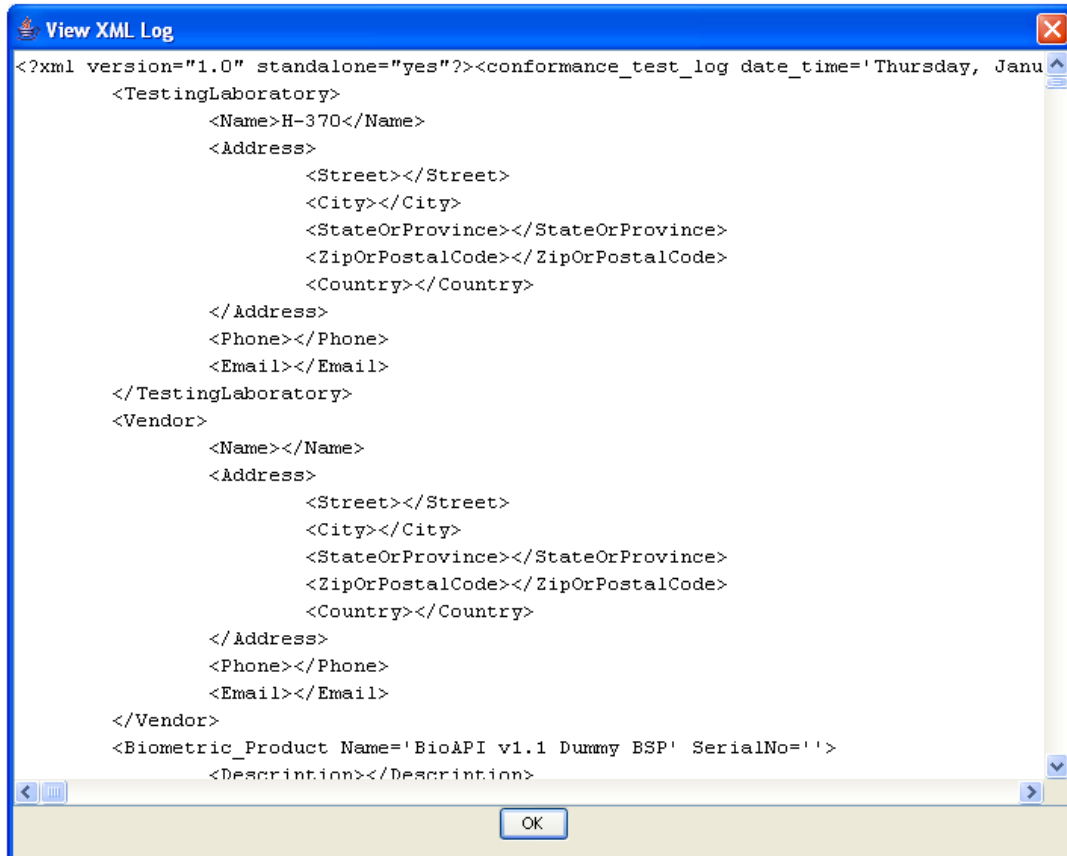
Click this button to change the Product Name to the Selected BSP Name and continue running the selected assertions. This is the recommended option as it ensures the XML log file will contain the correct BSP name.

**BTE Timeout Values**



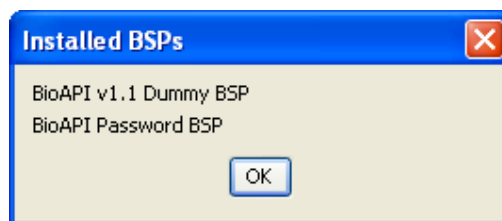
Use this dialog to change the timeout values used by the assertions. To display this dialog select *Timeout Values* from the **File** menu.

## View XML Log



To display the XML Log select the *View Log File* option on the **Test Log** menu.

## Installed BSPs



To display the list of installed BSP select the *Show Installed BSPs* option from the **Help** menu.

## Special Features

### Controlling Displayed BSPs

Sometimes, such as performing demos at a trade show, you may want to limit the BSPs shown in the Select BSP dropdown box. To do this follow these steps.

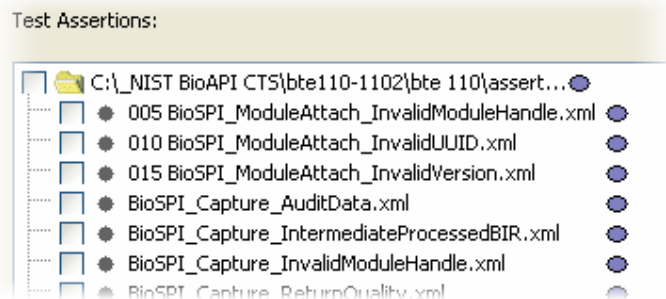
1. Create a file named  
    \_WhiteListBSPs.txt  
in the BTE folder. (This is the same folder where **run.bat** is located.)
2. Enter the BSPs you want displayed in the Select BSP dropdown box, one BSP per line. The names must exactly match, including upper and lower case.
3. Save the file.
4. Start the BTE and make sure the results are what you expect.

**Note:** All the installed BSPs will show up on the Installed BSPs dialog. This feature applies to the Select BSP dropdown box only.

### Controlling The Assertions Listed

Suppose you are a BSP vendor working on getting your BSP to pass all assertions. If you are working on just a few assertions it can be inconvenient to scroll through the entire list to select just the few you are interested in. There are two ways to increase your productivity.

1. Rename the assertions of interest and use the File Name Display Option. As shown here prepending numbers works well.





2. Create a folder under the **assertions** folder and move the assertions you do not want listed into the new folder. Be sure **not** to move the file **Auxiliary\_package.xml** as it is called by all assertions.

### LastLog.xml

After every assertion completes the BTE saves the complete XML Log to the file named

\_LastLog.xml

in the BTE folder. (The same folder where **run.bat** is located.)

This feature was added after a tester lost about a half hour of results when an errant BSP crashed the BTE.

### Message Boxes

If you want to display a message box before an assertion runs, such as to prompt the tester to do something, follow these steps.

1. In the **message** folder under the main BTE folder create a text file named the same as the assertion, except the extension is **.msg** instead of **.xml**. For example if you want a message to display when

BioSPI\_Enroll\_Payload.xml

is run, create the file

BioSPI\_Enroll\_Payload.msg

in the **message** folder.

**Note:** Despite the extension it is just a text file. Use Notepad or a similar program to create and edit the file.

2. Enter the text you want displayed in the file and save.

3. Run the BTE.

For an example of how this works remove the leading underscore from the sample file in the **message** folder.

### Supported Functions

Following is a list of currently supported BioAPI and BioSPI functions:

<b>BioSPI Functions</b>	
<b>Function</b>	<b>Brief Description</b>

<b>BioSPI Functions</b>	
<b>Function</b>	<b>Brief Description</b>
BioAPI_EnumModules	Enumerates the list of installed BSPs, and fills in BspSchemaArray with the results. If BspSchemaArray is NULL, simply counts the installed BSPs and returns the number in ElementsNeeded.
BioAPI_Init	This function initializes the BioAPI and verifies that the version of the BioAPI expected by the application is compatible with the version of the BioAPI on the system. This function should be called at least once by the application.
BioAPI_Terminate	This function terminates the caller's use of BioAPI. BioAPI can cleanup all internal state associated with the calling application. This function must be called once for each call to BioAPI_Init().
BioSPI_Capture	If this function is supported, Verification BSPs need only accept 'Purpose' flags indicating 'Verification' or 'Enroll for Verification'. If another purpose is set by the application, an error condition may be set as a BioAPI_RETURN. Similarly, this function need only return 'CapturedBIR' with the Purpose mask set to Verification or Enroll for Verification.
BioSPI_CreateTemplate	<b>BioAPI_RETURN BioAPI BioSPI_CreateTemplate</b> (BioAPI_HANDLE ModuleHandle, const BioAPI_INPUT_BIR *CapturedBIR, const BioAPI_INPUT_BIR *StoredTemplate, BioAPI_BIR_HANDLE_PTR NewTemplate, const BioAPI_DATA *Payload);
BioSPI_EnableEvents	<b>BioAPI_RETURN BioAPI BioSPI_EnableEvents</b> (BioAPI_HANDLE ModuleHandle, BioAPI_MODULE_EVENT_MASK *Events);
BioSPI_Enroll	<b>BioAPI_RETURN BioAPI BioSPI_Enroll</b> (BioAPI_HANDLE ModuleHandle, BioAPI_BIR_PURPOSE Purpose, const BioAPI_INPUT_BIR *StoredTemplate, BioAPI_BIR_HANDLE_PTR NewTemplate, const BioAPI_DATA *Payload, sint32 Timeout BioAPI_BIR_HANDLE_PTR AuditData);
BioSPI_FreeBirHandle	<b>BioAPI_RETURN BioAPI BioSPI_FreeBIRHandle</b>

<b>BioSPI Functions</b>	
<b>Function</b>	<b>Brief Description</b>
	(BioAPI_HANDLE ModuleHandle, BioAPI_BIR_HANDLE Handle);
BioSPI_GetBirFromHandle	<b>BioAPI_RETURN BioAPI BioSPI_GetBIRFromHandle</b> (BioAPI_HANDLE ModuleHandle, BioAPI_BIR_HANDLE Handle, BioAPI_BIR_PTR BIR);
BioSPI_GetHeaderFromHandle	<b>BioAPI_RETURN BioAPI BioSPI_GetHeaderFromHandle</b> (BioAPI_HANDLE ModuleHandle, BioAPI_BIR_HANDLE Handle, BioAPI_BIR_HEADER_PTR Header);
BioSPI_ModuleAttach	This function is invoked by BioAPI once for each invocation of BioAPI_ModuleAttach specifying the module identified by <i>BSPUuid</i> .  The service module must verify compatibility with the system version level specified by <i>Version</i> . If the version is not compatible, then this function fails. The service module should perform all initializations required to support the new attached session and should return a function table for the SPI entry points that can be invoked by BioAPI in response to API invocations. BioAPI uses this function table to dispatch requests on for the attach session created by this function. Each attach session has its own function table.
BioSPI_ModuleDetach	This function is invoked by BioAPI once for each invocation of BioAPI_ModuleDetach specifying the attach-session identified by <i>ModuleHandle</i> . The function entry point for <i>BioSPI_ModuleDetach</i> is included in the module function table BioAPI_MODULE_FUNCS returned to BioAPI as output of a successful <i>BioSPI_ModuleAttach</i> .
BioSPI_ModuleLoad	This function completes the module initialization process between BioAPI and the biometric service module.  The <i>BSPUuid</i> identifies the invoked module and should be used by the module to locate its entry in the module directory.
BioSPI_ModuleUnload	This function disables events and de-registers the BioAPI event-notification function. The biometric service module may perform cleanup operations, reversing the initialization performed in <i>BioSPI_ModuleLoad</i> .

<b>BioSPI Functions</b>	
<b>Function</b>	<b>Brief Description</b>
BioSPI_Process	<b>BioAPI_RETURN BioAPI BioSPI_Process</b> (BioAPI_HANDLE ModuleHandle, const BioAPI_INPUT_BIR *CapturedBIR, BioAPI_BIR_HANDLE_PTR ProcessedBIR);
BioSPI_Verify	<b>BioAPI_RETURN BioAPI BioSPI_Verify</b> (BioAPI_HANDLE ModuleHandle, const BioAPI_FAR *MaxFARRequested, const BioAPI_FRR *MaxFRRRequested, const BioAPI_BOOL *FARPrecedence, const BioAPI_INPUT_BIR *StoredTemplate, BioAPI_BIR_HANDLE_PTR AdaptedBIR, BioAPI_BOOL *Result, BioAPI_FAR_PTR FARAchieved, BioAPI_FRR_PTR FRRAchieved, BioAPI_DATA_PTR Payload, sint32 Timeout, BioAPI_BIR_HANDLE_PTR AuditData);
BioSPI_VerifyMatch	<b>BioAPI_RETURN BioAPI BioSPI_VerifyMatch</b> (BioAPI_HANDLE ModuleHandle, const BioAPI_FAR *MaxFARRequested, const BioAPI_FRR *MaxFRRRequested, const BioAPI_BOOL *FARPrecedence, const BioAPI_INPUT_BIR *ProcessedBIR, const BioAPI_INPUT_BIR *StoredTemplate, BioAPI_BIR_HANDLE_PTR *AdaptedBIR, BioAPI_BOOL *Result, BioAPI_FAR_PTR FARAchieved, BioAPI_FRR_PTR FRRAchieved, BioAPI_DATA_PTR Payload);

### **Software Distribution - File List**

Following is a list of files distributed in the **BTE 1\_1.zip** file:

<b>Name</b>	<b>Type</b>
<b>The following files are installed in the folder of the user's choice; the default folder name is \bte 110</b>	
bioapi100.dll	Application Extension

<b>Name</b>	<b>Type</b>
bioapi.jpg	JPEG Image
bioapi_bin.exe	Application
BioAPICTSTestLogSchema.xsd	XML Schema Description
BioAPICTSTestLogSchema.xsx	XML Schema Description
BioApiJni.jar	Executable Jar File
bte100.jar	Executable Jar File
BTE 1.1 User Guide.pdf	Adobe Acrobat Document
BTEMonitor.exe	Application
BTEServer.exe	Application
black.gif	GIF Image
bte.properties	PROPERTIES File
ecs-1.4.2.jar	Executable Jar File
green.gif	GIF Image
purple.gif	GIF Image
red.gif	GIF Image
ReleaseNotes.html	HTML Document
run.bat	MS-DOS Batch File
RunAssertion\$1.class	Compiled Java

Name	Type
	Module
RunAssertion.bat	MS-DOS Batch File
RunAssertion.class	Compiled Java Module
RunAssertion.java	Java File
Rundebug.bat	MS-DOS Batch File
runVIP.bat	MS-DOS Batch File
setup_bioapi.bat	MS-DOS Batch File
StartTest.class	Compiled Java Module
StartTest.java	Java File
test.dtd	DTD File
test.properties	PROPERTIES File
Timeout.properties	PROPERTIES File
white.gif	GIF Image
xml4j.jar	Executable Jar File
Yellow.gif	GIF Image
<p><b>The following files will be installed in a folder titled “assertions”, under the \bte 110 or other user-named directory (\bte 110\assertions)</b></p>	
Auxiliary_package.xml	XML Document

<b>Name</b>	<b>Type</b>
BioSPI_Capture_AuditData.xml	XML Document
BioSPI_Capture_IntermediateProcessedBIR.xml	XML Document
BioSPI_Capture_InvalidModuleHandle.xml	XML Document
BioSPI_Capture_ReturnQuality.xml	XML Document
BioSPI_CreateTemplate_BIRHeaderQuality.xml	XML Document
BioSPI_CreateTemplate_InputBIRDataType.xml	XML Document
BioSPI_CreateTemplate_OutputBIRDataType.xml	XML Document
BioSPI_CreateTemplate_OutputBIRPurpose.xml	XML Document
BioSPI_CreateTemplate_PayloadSupported.xml	XML Document
BioSPI_CreateTemplate_Purpose.xml	XML Document
BioSPI_CreateTemplate_StoredTemplateUnchanged.xml	XML Document
BioSPI_EnableEvents_InvalidModuleHandle.xml	XML Document
BioSPI_EnableEvents_ValidParam.xml	XML Document
BioSPI_Enroll_AuditData.xml	XML Document
BioSPI_Enroll_BIRHeaderQuality.xml	XML Document
BioSPI_Enroll_Payload.xml	XML Document
BioSPI_Enroll_ValidParam.xml	XML Document
BioSPI_FreeBIRHandle_InvalidBIRHandle.xml	XML Document
BioSPI_FreeBIRHandle_InvalidModuleHandle.xml	XML Document
BioSPI_FreeBIRHandle_ValidParam.xml	XML Document

<b>Name</b>	<b>Type</b>
BioSPI_GetBIRFromHandle_InvalidBIRHandle.xml	XML Document
BioSPI_GetBIRFromHandle_InvalidModuleHandle.xml	XML Document
BioSPI_GetBIRFromHandle_ValidParam.xml	XML Document
BioSPI_GetHeaderFromHandle_BIRHandleNotFreed.xml	XML Document
BioSPI_GetHeaderfromHandle_InvalidBIRHandle.xml	XML Document
BioSPI_GetHeaderfromHandle_InvalidModuleHandle.xml	XML Document
BioSPI_GetHeaderFromHandle_ValidParam.xml	XML Document
BioSPI_ModuleAttach_InvalidModuleHandle.xml	XML Document
BioSPI_ModuleAttach_InvalidUUID.xml	XML Document
BioSPI_ModuleAttach_InvalidVersion.xml	XML Document
BioSPI_ModuleAttach_ValidParam.xml	XML Document
BioSPI_ModuleDetach_Confirm.xml	XML Document
BioSPI_ModuleDetach_InvalidModuleHandle.xml	XML Document
BioSPI_ModuleDetach_ValidParam.xml	XML Document
BioSPI_ModuleLoad_InvalidUUID.xml	XML Document
BioSPI_ModuleLoad_ValidParam.xml	XML Document
BioSPI_ModuleUnload_Confirm.xml	XML Document
BioSPI_ModuleUnload_InvalidUUID.xml	XML Document
BioSPI_ModuleUnload_Unmatch.xml	XML Document
BioSPI_ModuleUnload_ValidParam.xml	XML Document



<b>Name</b>	<b>Type</b>
BioSPI_Process_BIRHeaderQuality.xml	XML Document
BioSPI_Process_BuildsProcessedBIR.xml	XML Document
BioSPI_Process_InputBIRDataType.xml	XML Document
BioSPI_Process_OutputBIRPurpose.xml	XML Document
BioSPI_Process_ValidParam.xml	XML Document
BioSPI_VerifyMatch_AchievedFRR.xml	XML Document
BioSPI_VerifyMatch_Adaptation.xml	XML Document
BioSPI_VerifyMatch_Inconsistent_Purpose.xml	XML Document
BioSPI_VerifyMatch_Payload.xml	XML Document
BioSPI_VerifyMatch_UnspecifiedFAR.xml	XML Document
BioSPI_VerifyMatch_ValidParam.xml	XML Document
BioSPI_Verify_AuditData.xml	XML Document
BioSPI_Verify_FARUnspecified.xml	XML Document
BioSPI_Verify_FRR.xml	XML Document
BioSPI_Verify_Payload.xml	XML Document
BioSPI_Verify_TemplateAdaptation.xml	XML Document
BioSPI_Verify_ValidParam.xml	XML Document