



BigFix[®] Enterprise Suite

Platform API Reference

BigFix, Inc.
Emeryville, CA

Last Modified: 9/2/2008
Version 7.1

© 1998–2008 BigFix, Inc. All rights reserved.

BigFix®, Fixlet® and "Fix it before it fails"® are registered trademarks of BigFix, Inc. i-prevention, Powered by BigFix, Relevance Engine, and related BigFix logos are trademarks of BigFix, Inc. All other product names, trade names, trademarks, and logos used in this documentation are the property of their respective owners. BigFix's use of any other company's trademarks, trade names, product names and logos or images of the same does not necessarily constitute: (1) an endorsement by such company of BigFix and its products, and (2) an endorsement of the company or its products by BigFix.

No part of this documentation may be reproduced, transmitted, or otherwise distributed in any form or by any means (electronic or otherwise) without the prior written consent of BigFix, Inc. You may not use this documentation for any purpose except in connection with your use or evaluation of BigFix software and any other use, including for reverse engineering such software or creating compatible software, is prohibited. If the license to the software that this documentation accompanies is terminated, you must immediately return this documentation to BigFix, Inc. and destroy all copies you may have.

All inquiries regarding the foregoing should be addressed to:

BigFix, Inc.
1480 64th Street
Emeryville, CA 94608

Contents

PREFACE	1
AUDIENCE	1
ORGANIZATION OF THIS MANUAL	1
CONVENTIONS USED IN THIS MANUAL	1
VERSIONS	1
INTRODUCTION	2
INSTALLATION & SETUP	3
INSTALLATION.....	3
API REFERENCE	4
BESAPI.XMLIMPORTER	4
BESAPI.ACTIONSTOPPER.....	6
BESAPI.RETRIEVEDPROPERTY	6
BESAPI.FIXLETMESSAGE	9
BESAPI.SITEMANAGER	13
BESAPI.ACTIONSITEMASTHEAD	14
BESAPI.SIGNINGKEYS	15
BESAPI.FIXLETACTIONCREATOR.....	16
BESAPI.FIXLETCREATOR.....	19
BESAPI.SETTINGSACTIONCREATOR	22
EXAMPLES	26
INDEX	32

Preface

Audience

This reference is for software developers who want to access the functionality of the BES Console from their own applications. These functions are exposed by the **BES Platform API**. This document serves as the external specification for the API and describes the relevant objects including their methods and properties.

Organization of this Manual

This document is organized as follows:

Introduction: This chapter contains a brief introduction to the BES Platform API.

Installation & Setup: This chapter describes the installation and setup procedures that need to be performed in order to use the BES Platform API.

API Reference: This chapter details the COM objects that collectively form the BES Platform API, including their methods, properties, and events.

Examples: This chapter contains an example illustrating the use of the API to perform an action from a script.

Conventions Used in this Manual

This document makes use of the following conventions and nomenclature:

Convention	Use
Bold Sans	A bold sans-serif font is used for API headers.
Mono-space	A mono-spaced font is used to indicate scripts or code snippets.

Versions

The document describes functionality in BES Version 7.1 and later.

Introduction

This document describes the Platform API for the BigFix Enterprise Suite (BES). The BES allows operators to maintain and audit hundreds of thousands of networked computers. It includes tools for the creation of custom Tasks and Actions, giving administrators a wide range of options for controlling and monitoring their networks.

The BES Console is the command and control interface that enables authorized users to perform these functions *interactively*. The BES Platform API allows external applications to perform these very same functions *programmatically*. Examples of such applications include customer-written scripts, automated operators, and third-party management applications/platforms that need to integrate with BES.

The BES Platform API consists of a set of COM objects distributed in the form of a DLL file. The API was created using ATL and C++ but can be accessed via COM from a variety of languages and scripting environments such as VBScript, JScript, and C#.

To use the API, the calling application creates a BESAPI object with an associated set of methods and properties. There are different BESAPI objects for each kind of task one might undertake using BES, such as creating an action or task (BESAPI.XMLImporter) or managing site subscriptions (BESAPI.SiteManager). For a detailed description of the XML format used by the BESAPI.XMLImporter object, including the structures and types of specific elements and attributes, please refer to the ***BES Import-Export Reference*** document.

In addition to the BES Platform API, applications can also query the database directly using the BES Database API. The database interface is provided primarily for reporting purposes, such as web-page creation, and consists of a set of SQL views that applications can access using ADO (or other ODBC or MSSQL compatible interfaces). Please refer to the ***BES Database API Reference*** document for details.

Installation & Setup

Installation

The BESAPI can be installed by the supplied BESAPI Installer program (setup.exe) on any computer that meets the system requirements for the BES Console. Before you can install the BESAPI, you must have obtained the site certificate from BigFix, created the **action site masthead** file, and installed the BES Server. You may also want to verify that the BES Server is functioning properly by using the BES Diagnostics Tool. The BES Administrator's Guide provides details on each of these steps.

To install the BESAPI:

1. Copy the **action site masthead** file into the BESAPI Installer (setup.exe) directory.
2. Run **setup.exe** and follow the instructions.

The BESAPI needs access to the signing keys of the specified user for most of its functionality. It finds those keys by looking for them in a registered location. To set the location of the keys for a given user, you must use the BESAPI.SigningKeys object documented below. You may notice that all of the BESAPI objects allow you to specify an application name and a DSN name. These default to "BESAPI" and "BES_bfenterprise" and there is usually no need to change them from their default values. If you do change them you will need to set up registry keys for the new application name, and/or configure the ODBC DSN manually. The BESAPI installer sets up the necessary keys and ODBC DSN to the default values.

API Reference

The objects in the BES Platform API are listed below and are detailed in the following sections:

- XMLImporter
- ActionStopper
- RetrievedProperty
- FixletMessage
- FixletCreator
- SettingsActionCreator
- SiteManager
- ActionSiteMasthead
- SigningKeys

The following objects are deprecated. They are still available in the API to provide for backward compatibility with consumers of the API prior to version 6.0. These objects will probably be removed in a future version of BES. Consumers of the API should use the XMLImporter object instead.

- FixletActionCreator
- FixletCreator
- SettingsActionCreator

BESAPI.XMLImporter

See the BES Import-Export Reference document for a detailed description of the XML format. Note that even though the console allows you to import BES XML documents with computer groups in them, the BESAPI.XMLImporter object does not support the creation of computer groups.

XMLImporter Methods	Description
<code>HRESULT SetAppName (BSTR appName) ;</code>	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
<code>HRESULT SetDSN (BSTR dsn) ;</code>	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.

XMLImporter Methods	Description
<pre>HRESULT ImportFile (BSTR xmlFilePath, BSTR customSiteName, BSTR username, BSTR password, [out, retval] VARIANT *ids);</pre>	<p>Connects to the database using the username and password. Reads the contents of the specified xml file, and imports it into the database, using the specified custom site name, or if that is the empty string, then the operator's action site, or the master action site, if the operator is a master operator.¹ Returns an array containing the database ids of the objects created.</p>
<pre>HRESULT Import (SAFEARRAY(byte) xml, BSTR customSiteName, BSTR username, BSTR password, [out, retval] VARIANT *ids);</pre>	<p>Connects to the database using the username and password. Imports the xml specified into the database, using the specified custom site name, or if that is the empty string, then the operator's action site, or the master action site, if the operator is a master operator.¹ Returns an array containing the database ids of the objects created.</p>
<pre>HRESULT ImportAction (SAFEARRAY(byte) actionXML, SAFEARRAY(byte) targetXML, UINT sourceSiteID, UINT sourceFixletID, BSTR username, BSTR password, [out, retval] long *id);</pre>	<p>Connects to the database using the username and password. Imports the action specified by the actionXML, using the targetXML for targeting, and using the specified source information. The appropriate site (operator site or master action site) is propagated depending on whether the username refers to a master operator or not. Returns an integer containing the database id of the action object created.</p>

XMLImporter Properties	Description
<pre>HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);</pre>	<p>If a method fails, this property returns a string containing a diagnostic message.</p>
<pre>HRESULT StatusMessage ([out, retval] BSTR *pVal);</pre>	<p>Returns a string describing the status of the current operation (e.g. propagation).</p>
<pre>HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);</pre>	<p>This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.</p>

¹ Note that only one site is propagated, so importing more than one object, or using the "auto activate analysis" feature may result in an object being created in the database without a propagation being performed to send that object to the BES agents.

BESAPI.ActionStopper

ActionStopper Methods	Description
<code>HRESULT SetAppName (BSTR appName);</code>	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
<code>HRESULT SetDSN (BSTR dsn);</code>	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.
<code>HRESULT StopAction (UINT actionID, BSTR dbUserName, BSTR password);</code>	Stops the existing action in the database with the given ID number, and uses the username and password to propagate a new version of the action site.
<code>HRESULT DeleteAction (UINT actionID, BSTR dbUserName, BSTR password);</code>	Permanently deletes the existing action in the database with the given ID number, and uses the username and password to propagate a new version of the action site. The action must be stopped or expired. Open actions cannot be deleted.

ActionStopper Properties	Description
<code>HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);</code>	If a method fails, this property returns a string containing a diagnostic message.
<code>HRESULT StatusMessage ([out, retval] BSTR *pVal);</code>	Returns a string describing the status of the current operation (e.g. propagation).
<code>HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);</code>	This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.

BESAPI.RetrievedProperty

RetrievedProperty Methods	Description
<code>HRESULT SetAppName (BSTR appName);</code>	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.

RetrievedProperty Methods	Description
HRESULT SetDSN (BSTR dsn);	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.
HRESULT Create (BSTR name, BSTR relevance, UINT delaySeconds, BSTR dbUserName, BSTR password);	Create a new retrieved property with the given name, relevance, and evaluation period. Use the username and password to propagate a new version of the action site. The ID property is set to the database ID of the newly created retrieved property. Note that this function will fail if a retrieved property with the same name already exists.
HRESULT Modify (long ID, BSTR name, BSTR relevance, BSTR dbUserName, BSTR password);	Modify the existing retrieved property with the given database ID to have the given name, relevance and evaluation period. Use the username and password to propagate a new version of the action site. Note that this function will fail if the name is changed and a retrieved property with the same name already exists.
HRESULT Delete (long ID, BSTR dbUserName, BSTR password);	Delete the existing retrieved property with the given database ID, and use the username and password to propagate a new version of the action site.
HRESULT Load (long ID, BSTR dbUserName, BSTR password);	Connect to the database using the given username and password, and set the ID, Name, and Relevance properties using the values stored in the database for the existing retrieved property with the given database ID.

RetrievedProperty Properties	Description
HRESULT ID ([out, retval] long *pVal);	The database ID number of the retrieved property. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT Name ([out, retval] BSTR *pVal);	The name of the retrieved property. Not set until after one of Load, Modify or Create has succeeded.
HRESULT Relevance ([out, retval] BSTR *pVal);	The relevance expression that defines the retrieved property. Not set until after one of Load, Modify or Create has succeeded.
HRESULT DelaySeconds ([out, retval] UINT *pVal);	The evaluation period of this retrieved property in seconds (0 means 'every report'). Not set until after one of Load, Modify, Delete or Create has succeeded.

RetrievedProperty Properties	Description
HRESULT IsCustom ([out, retval] BOOL *pVal);	True if the property is a custom property. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT IsReserved ([out, retval] BOOL *pVal);	True if the property is a reserved property. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT IsDefault ([out, retval] BOOL *pVal);	True if the property is a predefined property. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT IsPropertyOverride ([out, retval] BOOL *pVal);	True if the property is a reference to a property defined in an analysis. The evaluation period of this object overrides the evaluation period specified in the analysis. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT SourceSiteID ([out, retval] UINT *pVal);	If the property is a reference to a property defined in an analysis, then this is the SiteID of the site which contains the analysis. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT SourceFixletID ([out, retval] UINT *pVal);	If the property is a reference to a property defined in an analysis, then this is the FixletID of the analysis. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT SourcePropertyID ([out, retval] UINT *pVal);	If the property is a reference to a property defined in an analysis, then this is the ID of the property withing that analysis. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT SourceName ([out, retval] BSTR *pVal);	If the property is a reference to a property defined in an analysis, then this is the original name of the property in the analysis. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT SourceRelevance ([out, retval] BSTR *pVal);	If the property is a reference to a property defined in an analysis, then this is the original relevance of the property in the analysis. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT SourceDelaySeconds ([out, retval] BSTR *pVal);	If the property is a reference to a property defined in an analysis, then this is the original evaluation period for the property in the analysis. Not set until after one of Load, Modify, Delete or Create has succeeded.
HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);	If a method fails, this property returns a string containing a diagnostic message.

RetrievedProperty Properties	Description
HRESULT StatusMessage ([out, retval] BSTR *pVal);	Returns a string describing the status of the current operation (e.g. propagation).
HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);	This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.

BESAPI.FixletMessage

FixletMessage Methods	Description
HRESULT SetAppName (BSTR appName);	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
HRESULT SetDSN (BSTR dsn);	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.
HRESULT Load (long siteID, long fixletID, BSTR dbUserName, BSTR password);	Connect to the database using the given username and password. Query the database for the data of the fixlet message with the given fixletID contained in the fixlet site identified by the given siteID. The properties of the FixletMessage object will be set using the data read from the database.
HRESULT Delete (BSTR dbUserName, BSTR password);	Connect to the database using the given username and password. Deletes this fixlet, analysis, task, or computer group, and propagates the action site. This method fails if the Load method has not been used to load the object.
HRESULT Activate (BSTR dbUserName, BSTR password);	Connect to the database using the given username and password. Activates this analysis and propagates the action site. This method fails if the Load method has not been used to load the object, or if the object is not an analysis.
HRESULT Deactivate (BSTR dbUserName, BSTR password);	Connect to the database using the given username and password. Stops this analysis and propagates the action site. This method fails if the Load method has not been used to load the object, or if the object is not an analysis.

FixletMessage Properties	Description
HRESULT SiteID ([out, retval] long *pVal);	The database ID number of the site containing the loaded fixlet message. Not set until after Load has succeeded.
HRESULT FixletID ([out, retval] long *pVal);	The database ID number of the loaded fixlet message. Not set until after Load has succeeded.
HRESULT SiteName ([out, retval] long *pVal);	The name of the site containing the loaded fixlet message. Not set until after Load has succeeded.
HRESULT SiteDisplayName([out, retval] BSTR *pVal);	Version 7.1 introduces a new feature where content can rename a site, such that the site name in all the Console and Web Reports UI is shown with the new name. To maintain backward compatibility, the SiteName property (above) will still return the old site name, while this method will return the new site display name.
HRESULT SiteURL ([out, retval] long *pVal);	The gather URL of the site containing the loaded fixlet message. Not set until after Load has succeeded.
HRESULT Name ([out, retval] BSTR *pVal);	The name of the loaded fixlet message. Not set until after Load has succeeded.
HRESULT Relevance ([out, retval] BSTR *pVal);	The relevance expression, including parent relevance, which defines when the problem identified by this fixlet message affects a client. Not set until after Load has succeeded.
HRESULT ActionScript (long whichAction, [out, retval] BSTR *pVal);	The action script which corrects the problem identified by this fixlet message. Because the fixlet message may provide more than one corrective action, you must specify the whichAction parameter. Usually the whichAction parameter is 1. Not set until after Load has succeeded.
HRESULT ActionScriptMIMETYPE (long whichAction, [out, retval] BSTR *pVal);	The MIME type of the action script which corrects the problem identified by this fixlet message. Because the fixlet message may provide more than one corrective action, you must specify the whichAction parameter. Usually the whichAction parameter is 1. Not set until after Load has succeeded.

FixletMessage Properties	Description
HRESULT ActionScriptTypeName (long whichAction, [out, retval] BSTR *pVal);	The friendly name of the MIME type of the action script which corrects the problem identified by this fixlet message. Because the fixlet message may provide more than one corrective action, you must specify the whichAction parameter. Usually the whichAction parameter is 1. Not set until after Load has succeeded.
HRESULT HTML ([out, retval] BSTR *pVal);	The HTML code for the body of the fixlet message which describes the problem and offers links to the corrective actions. This HTML code may reference image files from the site. Those references will be relative, and can be resolved if you set the BASE property of the HTML document to the path to the site's data directory. Not set until after Load has succeeded.
HRESULT IsTask ([out, retval] BOOL *pVal);	True if the fixlet is a task message. Not set until after Load has succeeded.
HRESULT IsAnalysis ([out, retval] BOOL *pVal);	True if the fixlet is an analysis. Not set until after Load has succeeded.
HRESULT IsPlainFixlet ([out, retval] BOOL *pVal);	True if the fixlet is neither a task, analysis, baseline nor computer group. Not set until after Load has succeeded.
HRESULT IsBaseline ([out, retval] BOOL *pVal);	True if the fixlet is a baseline. Not set until after Load has succeeded.
HRESULT DownloadSize ([out, retval] UINT *pVal);	The size of the download for this fixlet, in bytes. Not set until after Load has succeeded.
HRESULT Source ([out, retval] BSTR *pVal);	A string describing the source of this fixlet message (e.g. Microsoft). Not set until after Load has succeeded.
HRESULT SourceID ([out, retval] BSTR *pVal);	A string describing an identifier specified by the source of the fixlet message (e.g. Microsoft KB number). Not set until after Load has succeeded.
HRESULT SourceSeverity ([out, retval] BSTR *pVal);	A string describing the severity rating of the fixlet message as determined by the source. Not set until after Load has succeeded.
HRESULT SourceReleaseDate ([out, retval] BSTR *pVal);	A string indicating when the source released the information. Not set until after Load has succeeded.

FixletMessage Properties	Description
HRESULT Category ([out, retval] BSTR *pVal);	A string describing the category of this fixlet as specified by the fixlet author. Not set until after Load has succeeded.
HRESULT Message ([out, retval] BSTR *pVal);	A string containing the text message specified when this fixlet message was created (only applies to custom fixlets). Not set until after Load has succeeded.
HRESULT IsDeleted ([out, retval] BOOL *pVal);	True if this fixlet message has been deleted (only applies to custom fixlets). Not set until after Load has succeeded.
HRESULT PropertyIDSet ([out, retval] BSTR *pVal);	A string containing the list of database IDs of the properties that refer to the properties contained in this analysis. The IDs are separated by tabs. This property is only set for analyses. Not set until after Load has succeeded.
HRESULT IsGloballyActivated ([out, retval] BOOL *pVal);	True if this fixlet is an analysis and has been activated by a Master Operator. Not set until after Load has succeeded.
HRESULT IsGloballyActivated ([out, retval] BOOL *pVal);	True if this fixlet is an analysis and has been activated by a Master Operator. Not set until after Load has succeeded.
HRESULT IsLocallyActivated ([out, retval] BOOL *pVal);	True if this fixlet is an analysis and has been activated by a Non-Master Operator. Not set until after Load has succeeded.
HRESULT CanActivate ([out, retval] BOOL *pVal);	True if this fixlet is an analysis and can be activated by the current user (the user which was used to Load this fixlet). Not set until after Load has succeeded.
HRESULT CanDeactivate ([out, retval] BOOL *pVal);	True if this fixlet is an analysis and can be stopped by the current user (the user which was used to Load this fixlet). Not set until after Load has succeeded.
HRESULT ActionXML (long whichAction, SAFEARRAY(byte) settingsXML, [out, retval] IXMLDOMDocument **actionXML);	Returns an importable XML document representing an action taken from this fixlet, task or baseline. Specify which action using the whichAction parameter. The settingsXML parameter is the bytes of an XML document that specifies the various settings in an action. See the BES Import-Export Reference document for more information.
HRESULT XML ([out, retval] IXMLDOMDocument **xml);	Returns an importable XML document representing this fixlet object.

FixletMessage Properties	Description
HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);	If a method fails, this property returns a string containing a diagnostic message.
HRESULT StatusMessage ([out, retval] BSTR *pVal);	Returns a string describing the status of the current operation (e.g. propagation).
HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);	This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.

BESAPI.SiteManager

SiteManager Methods	Description
HRESULT SetAppName (BSTR appName);	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
HRESULT SetDSN (BSTR dsn);	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.
HRESULT Subscribe (BSTR mastheadFilePath, BSTR username, BSTR password);	Connects to the database using the username and password. Subscribes to the site specified by the masthead file at the given file path. Propagates a new version of the action site.
HRESULT SubscribeWithRelevance (BSTR mastheadFilePath, BSTR relevance, BSTR username, BSTR password);	Connects to the database using the username and password. Subscribes computers that evaluate the specified relevance as “true” to the site specified by the masthead file at the given file path. Propagates a new version of the action site.
HRESULT Unsubscribe (UINT siteID, BSTR username, BSTR password);	Connects to the database using the username and password. Unsubscribes from the site specified by the given site ID. Propagates a new version of the action site.

SiteManager Properties	Description
HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);	If a method fails, this property returns a string containing a diagnostic message.

SiteManager Properties	Description
HRESULT StatusMessage ([out, retval] BSTR *pVal);	Returns a string describing the status of the current operation (e.g. propagation).
HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);	This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.

BESAPI.ActionSiteMasthead

ActionSiteMasthead Methods	Description
HRESULT SetAppName (BSTR appName);	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
HRESULT SetDSN (BSTR dsname);	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.
HRESULT Load (BSTR username, BSTR password);	Connect to the database using the given username and password. Read the action site masthead from the database.

ActionSiteMasthead Properties	Description
HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);	If a method fails, this property returns a string containing a diagnostic message.
HRESULT StatusMessage ([out, retval] BSTR *pVal);	Returns a string describing the status of the current operation (e.g. propagation).
HRESULT IsAdministrator(BSTR username, [out, retval] BOOL *pVal);	True if the operator specified is a Master Operator.
HRESULT IsAuthoringEnabled([out, retval] BOOL *pVal);	True if the authoring features are enabled for this BES deployment.

ActionSiteMasthead Properties	Description
<pre>HRESULT CurrentFIPSMoDe(BSTR username, BSTR password, [out, retval] BOOL* pVal);</pre>	<p>This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.</p>

BESAPI.SigningKeys

SigningKeys Methods	Description
<pre>HRESULT SetAppName (BSTR appName);</pre>	<p>Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.</p>
<pre>HRESULT SetDSN (BSTR dsn);</pre>	<p>Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.</p>
<pre>HRESULT SetPrivateKeyPath (BSTR username, BSTR fullpath);</pre>	<p>Register the location of the private key file for the given user. The fullpath parameter should be the full path to the private key file, including the file name, which is usually named "publisher.pvk".</p>
<pre>HRESULT SetPublisherCertPath (BSTR username, BSTR fullpath);</pre>	<p>Register the location of the user's certificate file that corresponds to the private key. The fullpath parameter should be the full path to the certificate file, including the file name, which is usually named "publisher.crt".</p>
<pre>HRESULT SetLicenseCertPath (BSTR username, BSTR fullpath);</pre>	<p>Register the location of the site license certificate file for the BES deployment. The fullpath parameter should be the full path to the certificate file, including the file name, which is usually named "license.crt".</p>
<pre>HRESULT SetDefaultAppName (BSTR appName);</pre>	<p>Set the default Application Name string which will be used by every BESAPI object, unless its SetAppName method is used to override the default. If no default is specified using this method and SetAppName is not called explicitly on an object, then the object will use 'BESAPI' as the application name.</p>

SigningKeys Methods	Description
<pre>HRESULT SetDefaultDSN (BSTR dsnName);</pre>	<p>Set the default name of the ODBC DSN to use when connecting to the database. This DSN will be used unless the SetDSN method is used to override the default. If no default is specified using this method, and SetDSN is not called explicitly on an object, then the object will use 'BES_bfenterprise' as the name of the ODBC DSN when connecting to the database.</p>

SigningKeys Properties	Description
<pre>HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);</pre>	<p>If a method fails, this property returns a string containing a diagnostic message.</p>
<pre>HRESULT StatusMessage ([out, retval] BSTR *pVal);</pre>	<p>Returns a string describing the status of the current operation (e.g. propagation).</p>
<pre>HRESULT AreSigningKeysValid(BSTR username, BSTR password, [out, retval] BOOL *pVal);</pre>	<p>True if the keys registered for the given user are valid, and the given password is correct. If this is false then the DiagnosticMessage property should specify the reason that the keys are not valid.</p>
<pre>HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);</pre>	<p>This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.</p>

BESAPI.FixletActionCreator

FixletActionCreator Methods	Description
<pre>HRESULT SetActionName (BSTR name);</pre>	<p>Specifies the name of the action, e.g. Action1.</p>
<pre>HRESULT SetTargetComputers (BSTR cidList);</pre>	<p>Specifies a list of computer IDs separated by characters.</p>
<pre>HRESULT SetTargetByList (BSTR listOfNames);</pre>	<p>Specifies a list of Computer Names, one per line.</p>

FixletActionCreator Methods	Description
HRESULT SetTargetRelevance (BSTR relevance);	Computers that evaluate the given expression to 'true' are targeted.
HRESULT SetStartTime (DATE start);	Action will only be taken after the given date and time.
HRESULT SetTimeRange (DATE start, DATE end);	Action will only be taken during the times of day in the given range (the dates given are ignored).
HRESULT SetActiveUserRequirement (BSTR requirement);	Valid argument values are 'User', 'NoUser', 'NoRequirement'.
HRESULT SetExpiration (DATE time);	Action will be expired after the given date and time.
HRESULT SetTryCount (UINT count);	Reapply the action up to this many times, if it becomes relevant again after having been successfully fixed (e.g. for policy enforcement).
HRESULT SetRetryCount (UINT count);	If the action fails, it will be retried up to this many times.
HRESULT SetRetryDelay (BSTR delay);	Amount of time to wait after failure before each retry attempt. String is of the form 'XX YY' where XX is a 1 or 2 digit number and YY is 'seconds', 'minutes', 'hours' or 'days'.
HRESULT SetRetryAfterReboot (BOOL retryAfterReboot);	After the action fails, wait for a system reboot before attempting to retry.
HRESULT SetReboot (BOOL reboot, BOOL allowCancel, BSTR title, BSTR text, UINT postponementSeconds, UINT forceRebootSeconds);	If reboot is true, BESClient will prompt the user and reboot after they click a button. If allowCancel is true then the dialog will contain a cancel button. If postponementSeconds is non-zero then the user can postpone up to that many seconds. If forceRebootSeconds is non-zero then reboot will be forced after that many seconds.

FixletActionCreator Methods	Description
<pre>HRESULT SetShutdown (BOOL shutdown, BOOL allowCancel, BSTR title, BSTR text, UINT postponementSeconds, UINT forceShutdownSeconds);</pre>	<p>If shutdown is true, BESClient will prompt the user and shutdown after they click a button. If allowCancel is true, the dialog will contain a cancel button. If postponementSeconds is non-zero, the user can postpone up to that many seconds. If forceShutdownSeconds is non-zero, shutdown will be forced after that many seconds.</p>
<pre>HRESULT SetTemporalDistribution(BSTR minutes);</pre>	<p>Clients will wait a random amount of time less than the given number of minutes before beginning to download the patch. The argument must be a number</p>
<pre>HRESULT SetMessage (BSTR title, BSTR text, BOOL details, BOOL cancel, UINT postponementSeconds, UINT timeoutSeconds);</pre>	<p>Before the action is run a dialog with the title and text (use CRLF line endings) will be displayed to the user. Depending on the flags, the dialog may include details and cancel buttons. If postponementSeconds is non-zero then the user can postpone up to that many seconds. If timeoutSeconds is non-zero then the message is automatically dismissed after that many seconds.</p>
<pre>HRESULT SetRunningMessage (BSTR title, BSTR text);</pre>	<p>While the action is running a dialog with the specified title and text (use CRLF line endings) will be displayed to the user.</p>
<pre>HRESULT AddParameter (BSTR name, BSTR value);</pre>	<p>If the action has a parameter with the given name, it will be assigned the given value.</p>
<pre>HRESULT SetFixletRelevance (BSTR relevance);</pre>	<p>The original fixlet's relevance. This expression must be true before the action can be applied, and the action will be successful when it becomes false.</p>
<pre>HRESULT SetActionScript (BSTR actionMIMEType, BSTR actionScript);</pre>	<p>Specify the action script for the action. This is usually copied from the fixlet message.</p>
<pre>HRESULT SetSourceInfo (UINT fixletID, BSTR fixletName, BSTR siteName, BSTR siteURL, UINT siteID);</pre>	<p>Specifies the name and identity of the fixlet message that the action is intended to fix.</p>

FixletActionCreator Methods	Description
HRESULT SetAppName (BSTR appName);	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
HRESULT SetDSN (BSTR dsn);	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.
HRESULT DoPropagation (BSTR dbUserName, BSTR password);	Commits the action to the Database and Propagates a new version of the ActionSite. Sets the ActionID property to the DB ID of the action created. If it fails, it sets the DiagnosticMessage property to a string value containing a description of the failure.

FixletActionCreator Properties	Description
HRESULT ActionID ([out, retval] long *pVal);	The database ID number of the propagated action. Not set until after DoPropagation has succeeded.
HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);	If DoPropagation fails (ActionID property is 0), then this property is a string that contains a diagnostic message that describes the failure.
HRESULT StatusMessage ([out, retval] BSTR *pVal);	Returns a string describing the status of the current propagation process. After initiating a propagation with the DoPropagation method, you can use this property to find out the current stage of the process.
HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);	This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.

BESAPI.FixletCreator

FixletCreator Methods	Description
HRESULT SetAppName (BSTR appName);	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
HRESULT SetDSN (BSTR dsn);	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.

FixletCreator Methods	Description
<code>HRESULT DoPropagation (BSTR username, BSTR password);</code>	Connects to the database using the username and password. Creates a fixlet using the values set in this object, and sets the FixletID property to the database ID of the newly created fixlet. Propagates a new version of the action site.
<code>HRESULT SetName (BSTR name);</code>	Set the name of the fixlet message, task, or analysis to be created. Default is 'Untitled'.
<code>HRESULT SetMessage (BSTR message);</code>	Specify the message displayed in the HTML body of the fixlet message, task, or analysis to be created.
<code>HRESULT SetCharset (BSTR charset);</code>	Specify the character set of the fixlet message, task, or analysis to be created. Default is 'us-ascii'.
<code>HRESULT SetRelevance (BSTR relevance);</code>	Specify the relevance expression for the fixlet message, task, or analysis to be created. Default is 'true'.
<code>HRESULT SetActionScript (BSTR actionScript);</code>	Specify the action script for the fixlet message, task, or analysis to be created. Note that custom analyses do not use an action script, but instead specify a PropertyIDSet using SetPropertyIDSet.
<code>HRESULT SetActionTypeMIME (BSTR actionTypeMIME);</code>	Specify the action script MIME type for the fixlet message, task, or analysis to be created. Default is 'application/x-fixlet-windows-shell'. Note that custom analyses do not use an action script, but instead specify a PropertyIDSet using SetPropertyIDSet.
<code>HRESULT SetActionDescription (BSTR actionDescription);</code>	Specify the description of the action for the fixlet message, task, or analysis to be created. This is displayed in the HTML body of the fixlet message and contains a hyperlink which when clicked initiates the action. The text between the first pair of underscores in the string provided will be turned into a hyperlink. Default is 'Click _here_ to deploy this action'.
<code>HRESULT SetCategory (BSTR category);</code>	Specify the category for the fixlet message, task, or analysis to be created.
<code>HRESULT SetDownloadSize (BSTR downloadSize);</code>	Specify the download size (in bytes) for the fixlet message, task, or analysis to be created.

FixletCreator Methods	Description
HRESULT SetSource (BSTR source);	Specify the source for the fixlet message, task, or analysis to be created.
HRESULT SetSourceID (BSTR sourceID);	Specify the source ID for the fixlet message, task, or analysis to be created.
HRESULT SetSourceSeverity (BSTR sourceSeverity);	Specify the source severity for the fixlet message, task, or analysis to be created.
HRESULT SetSourceReleaseDate (BSTR source);	Specify the source release date for the fixlet message, task, or analysis to be created.
HRESULT SetFixletType { BSTR type };	Specify the type of fixlet object. Must be one of “Fixlet”, “Task”, or “Analysis”. Default is “Fixlet”.
HRESULT SetIssuingFields (BSTR user, DATE time, BOOL isOperator);	Specify the issuing user, date and time issued, and a flag that is true if the user is a non-master operator at the time of issue.
HRESULT SetPropertyIDSet (BSTR propertyIDList);	Set the list of databaseIDs of the properties contained in this analysis. The IDs are separated by tab characters.
HRESULT SetActivateAnalysisFlag (BOOL activateAnalysis);	Set this flag to true in order to automatically activate the analysis after creating it.
HRESULT SetModifyFlag (BOOL modify);	Set this flag to true in order to modify an existing fixlet instead of creating a new one. The fixlet ID of the existing fixlet must be specified using SetFixletID.
HRESULT SetFixletID (UINT id);	If the modify flag is set to true, then this ID specifies which fixlet is being modified.

FixletCreator Properties	Description
HRESULT FixletID ([out, retval] long *pVal);	The database ID number of the propagated fixlet. Not set until after DoPropagation has succeeded.

FixletCreator Properties	Description
HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);	If a method fails, this property returns a string containing a diagnostic message.
HRESULT StatusMessage ([out, retval] BSTR *pVal);	Returns a string describing the status of the current operation (e.g. propagation).
HRESULT CurrentFIPSMODE(BSTR username, BSTR password, [out, retval] BOOL* pVal);	This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.

BESAPI.SettingsActionCreator

SettingsActionCreator Methods	Description
HRESULT SetAppName (BSTR appName);	Set the Application Name string, which is used to specify the registry key for the BESAPI. Default is 'BESAPI'.
HRESULT SetDSN (BSTR dsname);	Set the name of the ODBC DSN to use when connecting to the database. Default is 'BES_bfenterprise'.
HRESULT DoPropagation (BSTR username, BSTR password);	Connects to the database using the username and password. Creates a new settings action using the values set in this object, and sets the ActionID property to the database ID of the newly created settings action. Propagates a new version of the action site.
HRESULT SetActionName (BSTR name);	Specifies the name of the action, e.g. Action1.
HRESULT SetTargetComputers (BSTR cidList);	Specify a list of computer IDs separated by characters.
HRESULT SetTargetByList (BSTR listofNames);	Specify a list of Computer Names, one per line.
HRESULT SetTargetRelevance (BSTR relevance);	Computers that evaluate the given expression to 'true' are targeted.
HRESULT SetStartTime (DATE start);	Action will only be taken after the given date and time.

SettingsActionCreator Methods	Description
HRESULT SetTimeRange (DATE start, DATE end);	Action will only be taken during the times of day in the given range (the dates given are ignored).
HRESULT SetActiveUserRequirement (BSTR requirement);	Valid argument values are 'User', 'NoUser', 'NoRequirement'.
HRESULT SetExpiration (DATE time);	Action will be expired after the given date and time.
HRESULT SetTryCount (UINT count);	Reapply the action up to this many times, if it becomes relevant again after having been successfully fixed (e.g. for policy enforcement).
HRESULT SetRetryCount (UINT count);	If the action fails, it will be retried up to this many times.
HRESULT SetRetryDelay (BSTR delay);	Amount of time to wait after failure before each retry attempt. String is of the form 'XX YY' where XX is a 1 or 2 digit number and YY is 'seconds', 'minutes', 'hours' or 'days'.
HRESULT SetRetryAfterReboot (BOOL retryAfterReboot);	After the action fails, wait for a system reboot before attempting to retry.
HRESULT SetReboot (BOOL reboot, BOOL allowCancel, BSTR title, BSTR text, UINT postponementSeconds, UINT forceRebootSeconds);	If reboot is true, BESClient will prompt the user and reboot after they click a button. If allowCancel is true then the dialog will contain a cancel button. If postponementSeconds is non-zero then the user can postpone up to that many seconds. If forceRebootSeconds is non-zero then reboot will be forced after that many seconds.
HRESULT SetShutdown (BOOL shutdown, BOOL allowCancel, BSTR title, BSTR text, UINT postponementSeconds, UINT forceShutdownSeconds);	If shutdown is true, BESClient will prompt the user and shutdown after they click a button. If allowCancel is true then the dialog will contain a cancel button. If postponementSeconds is non-zero then the user can postpone up to that many seconds. If forceShutdownSeconds is non-zero then shutdown will be forced after that many seconds.
HRESULT SetTemporalDistribution (BSTR minutes);	Clients will wait a random amount of time less than the given number of minutes before beginning to download the patch. The argument must be a number.

SettingsActionCreator Methods	Description
<pre>HRESULT SetMessage (BSTR title, BSTR text, BOOL details, BOOL cancel, UINT postponementSeconds, UINT timeoutSeconds);</pre>	<p>Before the action is run a dialog with the title and text (use CRLF line endings) will be displayed to the user. Depending on the flags, the dialog may include details and cancel buttons. If postponementSeconds is non-zero then the user can postpone up to that many seconds. If timeoutSeconds is non-zero then the message is automatically dismissed after that many seconds.</p>
<pre>HRESULT SetRunningMessage (BSTR title, BSTR text);</pre>	<p>While the action is running a dialog with the specified title and text (use CRLF line endings) will be displayed to the user.</p>
<pre>HRESULT AddParameter (BSTR name, BSTR value);</pre>	<p>If the action has a parameter with the given name, it will be assigned the given value.</p>
<pre>HRESULT SetFixletRelevance (BSTR relevance);</pre>	<p>The original fixlet's relevance. This expression must be true before the action can be applied, and the action will be successful when it becomes false.</p>
<pre>HRESULT SetActionScript (BSTR actionMIMEType, BSTR actionScript);</pre>	<p>Specify the action script for the action. This is usually copied from the fixlet message.</p>
<pre>HRESULT SetSourceInfo (UINT fixletID, BSTR fixletName, BSTR siteName, BSTR siteURL, UINT siteID);</pre>	<p>Specifies the name and identity of the fixlet message that the action is intended to fix.</p>
<pre>HRESULT SetEffectiveDate (DATE time);</pre>	<p>Specify the effective date of the settings actions. If this date is older than the effective date of the current value of the setting on the target machine, then the action will not change the value of the setting.</p>
<pre>HRESULT AddIndefiniteLock();</pre>	<p>Add a line to the action script which locks the target machine indefinitely (a locked machine is prevented from running any Fixlet actions).</p>
<pre>HRESULT AddLock (DATE expiration);</pre>	<p>Add a line to the action script which locks the target machine until the given date and time (a locked machine is prevented from running any Fixlet actions).</p>

SettingsActionCreator Methods	Description
<code>HRESULT AddUnlock();</code>	Add a line to the action script which unlocks the target machine (a locked machine is prevented from running any Fixlet actions).
<code>HRESULT AddSetSetting (BSTR name, BSTR value, BOOL createRPFlag);</code>	Add a line to the action script which sets the named setting to the given value. The createRPFlag is currently unimplemented. If you require a retrieved property that returns the value of this setting you must create one using the BESAPI.RetrievedProperty object.
<code>HRESULT AddDeleteSetting (BSTR name);</code>	Add a line to the action script which deletes the value of the named setting.
<code>HRESULT AddSetManagementRights (BSTR operatorName, BSTR assignmentRelevance, UINT oldActionID);</code>	Management rights will be granted/revoked for the given operator if the given relevance expression evaluates to true/false. You must specify the action ID of the existing management rights action for the given operator, if there is one, so that it can be stopped (there should be only one Open management rights action for each operator).

SettingsActionCreator Properties	Description
<code>HRESULT ActionID ([out, retval] long *pVal);</code>	The database ID number of the propagated action. Not set until after DoPropagation has succeeded
<code>HRESULT DiagnosticMessage ([out, retval] BSTR *pVal);</code>	If DoPropagation fails (ActionID property is 0), then this property is a string that contains a diagnostic message that describes the failure.
<code>HRESULT StatusMessage ([out, retval] BSTR *pVal);</code>	Returns a string describing the status of the current propagation process. After initiating a propagation with the DoPropagation method, you can use this property to retrieve a string describing what stage of the process is currently underway.
<code>HRESULT CurrentFIPSMode(BSTR username, BSTR password, [out, retval] BOOL* pVal);</code>	This is a function added in version 7.1 to every BESAPI interface. It retrieves the masthead from the database and returns true if FIPS mode cryptography is enabled in the masthead.

Examples

To take an action from a script, first create a BESAPI.FixletActionCreator object.

For example:

```
var actionCreator = CreateObject("BESAPI.FixletActionCreator");
```

or

```
var actionCreator = new  
ActiveXObject("BESAPI.FixletActionCreator");
```

Then use the Set* methods on that object to specify the attributes of the action, such as which computers to target, when to run and the content of the action (i.e. when it is relevant, and the script to execute).

For example:

```
actionCreator.SetTargetComputers(" |12345678| ");
```

Finally, commit the action to the database and send it to the client computers using the DoPropagation method.

For example:

```
actionCreator.DoPropagation("joebob", "mypassword");  
window.alert("Propagated action id #" + actionCreator.actionID);
```

For more complete examples, see the sample code included in the TestAPI.htm file.

Here are some sample Perl scripts:

```
# BESAPI.pm  
package BESAPI;  
use strict;  
use Win32::OLE;  
  
Win32::OLE->Option( Warn => sub { die Win32::OLE->LastError() . "\n"; } );  
  
sub new  
{  
    my ( $perlType, $besAPIType ) = @_;  
    my $self = {};  
    $self->{type} = $besAPIType;  
    $self->{object} = Win32::OLE->new( "BESAPI." . $besAPIType );  
    return bless $self, $perlType;  
}
```

```
our $AUTOLOAD;
sub AUTOLOAD
{
    my $self = shift;
    $AUTOLOAD =~ s/^.*:://;
    my $besAPIObject = $self->{object};
    my $result = $self->{object}->Invoke( $AUTOLOAD, @_ );
    if ( $besAPIObject->DiagnosticMessage() )
    {
        die "BESAPI::" . $self->{type} . "::" . $AUTOLOAD . " Error: " . $self->{object}->DiagnosticMessage() . "\n";
    }

    return $result;
}

sub CheckSigningKeys
{
    my ( $dsn, $username, $password, $privateKey ) = @_;
    my $signingKeys = BESAPI->new( "SigningKeys" );
    $signingKeys->SetDefaultDSN( $dsn );
    $signingKeys->SetPrivateKeyPath( $username, $privateKey );
    if ( !$signingKeys->AreSigningKeysValid( $username, $password ) )
    { die "SigningKeys not valid\n"; }
}

sub FixletMessage
{
    my ( $siteID, $fixletID, $username, $password ) = @_;
    my $fixlet = BESAPI->new( "FixletMessage" );
    $fixlet->Load( $siteID, $fixletID, $username, $password );
    return $fixlet;
}

1;
```

The following script loads a FixletMessage with a specific site ID and fixlet ID and takes an action from that fixlet, using the FixletMessage.ActionXML and XMLImporter.ImportAction methods:

```
# TakeActionFromFixlet.pl

use strict;
use FindBin;
use lib $FindBin::Bin;
use BESAPI;

my $username = "bigfix";
my $password = "bigfix";
my $dsn      = "bes__";
my $privateKey = "\\10.10.42.39\\c\\$\\Keys\\bigfix\\publisher.pvk";

my $siteID    = 3096;
my $fixletID  = 3;
my $action    = 0;

BESAPI::CheckSigningKeys( $dsn, $username, $password, $privateKey );

my $fixlet = BESAPI::FixletMessage( $siteID, $fixletID, $username, $password );

my $settingsXML =
    "<?xml version='1.0'?'>\n" .
    "<ActionSettings>\n" .
    "  <Settings>\n" .
    "    <ActionUITitle>title</ActionUITitle>\n" .
    "    <PreActionShowUI>true</PreActionShowUI>\n" .
    "    <PreAction>\n" .
    "      <Text>preaction description</Text>\n" .
    "      <AskToSaveWork>true</AskToSaveWork>\n" .
    "      <ShowActionButton>true</ShowActionButton>\n" .
    "      <ShowCancelButton>true</ShowCancelButton>\n" .
    "      <DeadlineBehavior>ForceToRun</DeadlineBehavior>\n" .
    "      <DeadlineType>Absolute</DeadlineType>\n" .
    "      <DeadlineOffset>PT23H58M54.000000S</DeadlineOffset>\n" .
    "      <ShowConfirmation>true</ShowConfirmation>\n" .
    "      <Confirmation>confirmation message</Confirmation>\n" .
    "    </PreAction>\n" .
```

```
"<HasRunningMessage>true</HasRunningMessage>\n" .
"<RunningMessage>\n" .
    "<Text>running text</Text>\n" .
"</RunningMessage>\n" .
"<HasTimeRange>>false</HasTimeRange>\n" .
"<HasStartTime>true</HasStartTime>\n" .
"<StartDateTimeOffset>-PT1M6.000000S</StartDateTimeOffset>\n" .
"<HasEndTime>true</HasEndTime>\n" .
"<EndDateTimeOffset>P1DT23H58M54.000000S</EndDateTimeOffset>\n" .
"<HasDayOfWeekConstraint>>false</HasDayOfWeekConstraint>\n" .
"<ActiveUserRequirement>RequireUser</ActiveUserRequirement>\n" .
"<ActiveUserType>LocalUsers</ActiveUserType>\n" .
"<HasWhose>>false</HasWhose>\n" .
"<Reapply>>false</Reapply>\n" .
"<HasReapplyLimit>>false</HasReapplyLimit>\n" .
"<HasReapplyInterval>>false</HasReapplyInterval>\n" .
"<HasRetry>>false</HasRetry>\n" .
"<HasTemporalDistribution>>false</HasTemporalDistribution>\n" .
"<PostActionBehavior Behavior=\"Nothing\"></PostActionBehavior>\n" .
"<IsOffer>true</IsOffer>\n" .
"<OfferCategory>offer category</OfferCategory>\n" .
"<OfferDescriptionHTML>offer description</OfferDescriptionHTML>\n" .
"</Settings>" .
"</ActionSettings>" ;
```

```
my $actionXMLDocument = $fixlet->ActionXML( $action, $settingsXML );
```

```
my $targetXMLA =
```

```
"<?xml version=\"1.0\"?>" .
"<BESActionTarget>" .
    "<ComputerName>a</ComputerName>" .
    "<ComputerName>b</ComputerName>" .
    "<ComputerName>c</ComputerName>" .
"</BESActionTarget>" ;
```

```
my $targetXMLB =
```

```
"<?xml version=\"1.0\"?>" .
"<BESActionTarget>" .
    "<ComputerID>34</ComputerID>" .
```



```
"<ComputerID>12704810</ComputerID>" .
"</BESActionTarget>";

my $targetXMLC =
    "<?xml version=\"1.0\"?>" .
    "<BESActionTarget>" .
        "<CustomRelevance>exists file \"c:\\virus\"</CustomRelevance>" .
    "</BESActionTarget>";

my $xmlImporter = BESAPI->new( "XMLImporter" );
my $actionID = $xmlImporter->ImportAction( $actionXMLDocument->XML(), $targetXMLA,
$fixlet->SiteID(), $fixlet->FixletID(), $username, $password );

print "ActionID: $actionID\n";
```

This Perl script loads a FixletMessage with a specific site ID and Fixlet ID and prints out all the accessible information about that Fixlet:

```
# FixletMessage.pl

use strict;
use FindBin;
use lib $FindBin::Bin;
use BESAPI;

my $username = "bigfix";
my $password = "bigfix";
my $dsn      = "bes__";
my $privateKey = "\\10.10.42.21\\c$\\Keys\\bigfix\\publisher.pvk";

my $siteID = 1;
my $fixletID = 177;

BESAPI::CheckSigningKeys( $dsn, $username, $password, $privateKey );

my $fixlet = BESAPI::FixletMessage( $siteID, $fixletID, $username, $password );

print "Name: " . $fixlet->Name() . "\n";
print "SiteName: " . $fixlet->SiteName() . "\n";
print "SiteDisplayName: " . $fixlet->SiteDisplayName() . "\n";
print "SiteURL: " . $fixlet->SiteURL() . "\n";
```

EXAMPLES

```
print "SiteID: " . $fixlet->SiteID() . "\n";
print "FixletID: " . $fixlet->FixletID() . "\n";
print "IsTask: " . $fixlet->IsTask() . "\n";
print "IsAnalysis: " . $fixlet->IsAnalysis() . "\n";
print "IsPlainFixlet: " . $fixlet->IsPlainFixlet() . "\n";
print "IsBaseline: " . $fixlet->IsBaseline() . "\n";
print "DownloadSize: " . $fixlet->DownloadSize() . "\n";
print "Source: " . $fixlet->Source() . "\n";
print "SourceID: " . $fixlet->SourceID() . "\n";
print "SourceSeverity: " . $fixlet->SourceSeverity() . "\n";
print "SourceReleaseDate: " . $fixlet->SourceReleaseDate() . "\n";
print "Category: " . $fixlet->Category() . "\n";
print "IsDeleted: " . $fixlet->IsDeleted() . "\n";
print "Relevance: " . $fixlet->Relevance() . "\n";
print "Current FIPS Mode: " . $fixlet->CurrentFIPSMode( $username, $password );

if ( $fixlet->IsAnalysis() )
{
    print "PropertyIDSet: " . $fixlet->PropertyIDSet() . "\n";
    print "IsGloballyActivated: " . $fixlet->IsGloballyActivated() . "\n";
    print "IsLocallyActivated: " . $fixlet->IsLocallyActivated() . "\n";
    print "CanActivate: " . $fixlet->CanActivate() . "\n";
    print "CanDeactivate: " . $fixlet->CanDeactivate() . "\n";
}

print "\n\n";
print "Message: " . $fixlet->Message() . "\n";
print "HTML: \n\n" . $fixlet->HTML() . "\n";

if ( $fixlet->IsPlainFixlet() || $fixlet->IsTask() )
{
    my $action = 0;
    print "ActionScript: " . $fixlet->ActionScript( $action ) . "\n";
    print "ActionScriptMIMETYPE: " . $fixlet->ActionScriptMIMETYPE( $action ) . "\n";
    print "ActionScriptTypeName: " . $fixlet->ActionScriptTypeName( $action ) . "\n";
}
```

6INDEX

Index

A

action site masthead · 3, 14

ActionID · 19, 22, 25

ActionScript · 10

ActionScriptMimeType · 10

ActionScriptTypeName · 11

Activate · 9

AddDeleteSetting · 25

AddIndefiniteLock · 24

AddLock · 24

AddParameter · 18, 24

AddSetManagementRights · 25

AddSetSetting · 25

AddUnlock · 25

ADO · 2

AreSigningKeysValid · 16

Audience · 1

B

BES Administrator's Guide · 3

BES Console · 1, 2, 3

BES Database · 2

BES Database API · 2

BES Diagnostics Tool · 3

BES Server · 3

BES_bfenterprise · 3, 4, 6, 7, 9, 13, 14, 15, 19, 22

BESAPI object · 2

BESAPI.ActionSiteMasthead · 14

BESAPI.ActionStopper · 6

BESAPI.FixletActionCreator · 16

BESAPI.FixletCreator · 19

BESAPI.FixletMessage · 9

BESAPI.RetrievedProperty · 6

BESAPI.SettingsActionCreator · 22

BESAPI.SigningKeys · 15

BESAPI.SiteManager · 4, 13

C

CanActivate · 12

CanDeactivate · 12

Category · 12

COM · 1, 2

Conventions Used in this manual · 1

Create · 7, 8

CurrentFIPSMODE · 5, 6, 9, 13, 14, 15, 16, 19, 22, 25

D

Deactivate · 9

DelaySeconds · 7

Delete · 7, 8, 9

DiagnosticMessage · 5, 6, 8, 13, 14, 16, 19, 22, 25

DLL · 2

DoPropagation · 19, 20, 21, 22, 25, 26

DownloadSize · 11

DSN · 3, 4, 6, 7, 9, 13, 14, 15, 19, 22

F

FixletID · 8, 10, 20, 21

H

HTML · 11, 20

6INDEX

I

ID · 6, 7, 8, 10, 13, 19, 20, 21, 22, 25
Installation & Setup · 3
IsAdministrator · 14
IsAnalysis · 11
IsAuthoringEnabled · 14
IsCustom · 8
IsDefault · 8
IsDeleted · 12
IsGloballyActivated · 12
IsLocallyActivated · 12
IsPlainFixlet · 11, 12
IsPropertyOverride · 8
IsReserved · 8
IsTask · 11

L

Load · 7, 8, 9, 10, 11, 12, 14

M

Message · 12
methods · 1, 2, 26
Modify · 7, 8
MSSQL · 2

N

Name · 4, 6, 7, 9, 10, 13, 14, 15, 19, 22

O

ODBC · 2, 3, 4, 6, 7, 9, 13, 14, 15, 19, 22

P

PropertyIDSet · 12, 20

R

Relevance · 7, 10
Relevance Language · 1

S

SetActionDescription · 20
SetActionName · 16, 22
SetActionScript · 18, 20, 24
SetActionTypeMIME · 20
SetActivateAnalysisFlag · 21
SetActiveUserRequirement · 17, 23
SetAppName · 4, 6, 9, 13, 14, 15, 19, 22
SetCategory · 20
SetCharset · 20
SetDownloadSize · 20
SetDSN · 4, 6, 7, 9, 13, 14, 15, 19, 22
SetEffectiveDate · 24
SetExpiration · 17, 23
SetFixletID · 21
SetFixletRelevance · 18, 24
SetIssuingFields · 21
SetLicenseCertPath · 15
SetMessage · 18, 20, 24
SetModifyFlag · 21
SetName · 20
SetPrivateKeyPath · 15
SetPropertyIDSet · 20, 21
SetPublisherCertPath · 15
SetReboot · 17, 23
SetRelevance · 20
SetRetryAfterReboot · 17, 23
SetRetryCount · 17, 23
SetRetryDelay · 17, 23
SetRunningMessage · 18, 24

6INDEX

SetShutdown · 18, 23

SetSource · 21

SetSourceID · 21

SetSourceInfo · 18, 24

SetSourceReleaseDate · 21

SetSourceSeverity · 21

SetStartTime · 17, 22

SetTargetByList · 16, 22

SetTargetComputers · 16, 22, 26

SetTargetRelevance · 17, 22

SetTemporalDistribution · 18, 23

SetTimeRange · 17, 23

SetTryCount · 17, 23

setup.exe · 3

signing keys · 3

site certificate · 3

SiteDisplayName · 10

SiteID · 8, 10

SiteName · 10

SiteURL · 10

Source · 11

SourceDelaySeconds · 8

SourceFixletID · 8

SourceID · 11

SourceName · 8

SourcePropertyID · 8

SourceReleaseDate · 11

SourceRelevance · 8

SourceSeverity · 11

SourceSiteID · 8

StatusMessage · 5, 6, 9, 13, 14, 16, 19, 22, 25

Subscribe · 5, 13

U

Unsubscribe · 13