BigFix® Enterprise Suite (BES™)

Database API Reference

BigFix, Inc.
Emeryville, CA

Last Modified: 4/28/2005

Version 5.1

BigFix, Inc.
6121 Hollis Street
Emeryville, CA 94608-2021

Contents

Preface

Audience

This reference is for users of the BigFix Enterprise Suite (BES) who are developing applications that need to make customized queries against the BES SQL Database, such as for generating reports. The document describes a set of SQL views that constitutes the BES Database Application Programming Interface (API) that will help you write those applications.

Organization of this manual

This document is organized as follows:

- **Introduction.** This chapter contains a brief introduction to SQL and the BES Database API.
- **View Schema.** This chapter provides details of the views that comprise the BES Database API. For each view, a table is provided that lists the column names along with the corresponding data types and descriptions. Sample queries are also provided for many of the views.
- **Example Report Generation.** This chapter contains a simple Perl script that generates an HTML report from four of the views.

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 the names of views.
`Mono-space`	A mono-spaced font is used for sample programs and scripts.

Versions

The functionality discussed in this document was first introduced in the **Enterprise 1.3** database included in BES 4.0.0.1 and later. This version of the document describes the views included in the **Enterprise 1.55** database included in BES 5.1 and later.

Introduction

The BES Database API consists of a set of SQL views that ship with the BES SQL database. These views are provided to enable customer and third-party applications to query the database directly using MSSQL compatible interfaces such as ADO or ODBC. A typical application might be a Perl cgi program that creates an HTML report for online viewing. Perl uses the DataBase Interface (dbi) to connect to the SQL database. Any programming language that has an ODBC interface can be used to access the database.

The SQL format of the BES Database makes it easy to create various views of the tables, including Fixlet, Action, Computer and Retrieved Property tables. With a few simple SELECT commands, you can create filtered and sorted views of the various databases. These can be used to prepare custom reports, audit trails or to capture snapshots of the BES environment.

The BES Database API is intended to provide backwards compatibility across releases: applications written against them should continue to work in newer releases of BES unless product functionality or underlying content changes in a way that renders these views inapplicable. In future releases, BigFix may add additional columns to these views and introduce new views and stored procedures, but an attempt will be made to avoid removing any existing functionality.

Access to the database for these SELECT commands is granted to all authorized users of the BES Console. Since these views are intended for output only, users won't be able to update, delete or otherwise modify the database with this API. For information on how to create actions and tasks that may modify the BES Database, see the **BES Platform API** reference manual.

View Schema

The following sections describe each of the views provided by the BES SQL database.

BES_FIXLETS

The BES_FIXLETS view provides a list of all Fixlets in the BES Database. This table is useful for joining against the BES_RELEVANT_FIXLETS and BES_ACTIONS table to get the Fixlet name. Custom Fixlet content is provided under the "ActionSite" sitename.

Column	Type	Description
Sitename	varchar(128)	Fixlet site name
ID	int	Unique Fixlet ID
Name	varchar(255)	Fixlet name

Example:

- select Sitename, ID, Name from BES_Fixlets where Sitename = 'Enterprise Security' order by Sitename, ID

BES_TASKS

The BES_TASKS view provides a list of all Tasks (including custom Tasks) in the BES Database. This table is useful for joining against the BES_RELEVANT_TASKS and BES_ACTIONS table to get the Task name.

Column	Type	Description
Sitename	varchar(128)	Source Fixlet site name
ID	int	Unique Task ID
Name	varchar(255)	Task name

Example:

- select Sitename, ID, Name from BES_Tasks where Sitename = 'Enterprise Security' order by Sitename, ID

BES_ANALYSES

The BES_TASKS view provides a list of all Analyses (including custom Analyses) in the BES Database.

Column	Type	Description
Sitename	varchar(128)	Source Fixlet site name
ID	int	Unique Analysis ID
Name	varchar(255)	Analysis name

Example:

- select Sitename, ID, Name from BES_Analyses where Sitename = 'BES Support' order by Sitename, ID

BES_ COMPUTERGROUPS

The BES_COMPUTERGROUPS view provides a list of all Computer Groups in the BES Database.

Column	Type	Description
ID	int	Unique Group ID
Name	varchar(255)	Computer group name

Example:

- select ID, Name from BES_ComputerGroups where Name LIKE 'Chicago Office%' order by ID

BES_COLUMN_HEADINGS

The BES_COLUMN_HEADINGS view provides access to all the retrieved property information collected about client computers by the BES Database. Retrieved properties which return multiple results will be expressed in this view by a value field which contains the multiple results separated by a newline character. As a performance optimization in 5.1, column headings whose "Value" contains more than 8000 characters will be truncated to 8000 characters in this view.

Column	Type	Description

Column	Type	Description
ComputerID	int	Computer ID
Name	varchar(255)	Retrieved property name
Value	varchar(8000)	Newline separated list of retrieved property values
IsFailure	Tinyint	Non-zero if the retrieved property failed to evaluate on the BES Client

Example:

- select ComputerID, Name, Value, IsFailure from BES_COLUMN_HEADINGS where Name = 'Total HD Space' order by ComputerID

BES_RELEVANT_FIXLETS

The BES_RELEVANT_FIXLETS view contains an entry for every Fixlet/computer pair in which the Fixlet is relevant on that computer. This view has been modified in 5.1 to include custom Fixlet content.

Column	Type	Description
Sitename	varchar(128)	Fixlet site name
ID	int	Fixlet ID
ComputerID	int	Computer ID
Version	int	Number of times the Fixlet message has been modified

Example:

- select F.Sitename, F.ID, F.Name, R.ComputerID from BES_FIXLETS F, BES_RELEVANT_FIXLETS R where F.Sitename = R.Sitename AND F.ID = R.ID

BES_RELEVANT_TASKS

The BES_RELEVANT_TASKS view contains an entry for every Task/computer pair (including custom Tasks) in which the Task is relevant on that computer.

Column	Type	Description
Sitename	varchar(128)	Fixlet site name

Column	Type	Description
ID	int	Task ID
ComputerID	int	Computer ID
Version	int	Number of times the Task has been modified

Example:

- select T.Sitename, T.ID, T.Name, R.ComputerID from BES_TASKS T, BES_RELEVANT_TASKS R where T.Sitename = R.Sitename AND T.ID = R.ID

BES_ACTIONS

The BES_ACTIONS view contains an entry for every Action/computer pair where the action was received by the computer.

Column	Type	Description
ActionID	int	Action ID
ComputerID	int	Computer ID
Name	varchar(255)	Title of the action
Username	varchar(32)	Database user name of action issuer
StartTime	datetime	Time at which the action was issued
FixletID	int	Source Fixlet ID
Sitename	varchar(128)	Source Fixlet site name
ActionStatus	text	A brief summary of the state of the action for this computer

Example:

- select * from BES_ACTIONS where ActionStatus = ‘Executed’

BES_RELEVANT_FIXLET_HISTORY

The BES_RELEVANT_FIXLET_HISTORY view contains an entry for every Fixlet/Computer pair that has ever been relevant, with timestamps indicating the first time it became relevant, the last time it became relevant (the same as FirstBecameRelevant if it only became relevant once), and the last time it became non-relevant. Some of these fields may be NULL if the event in question never occurred or if it occurred before upgrading to the BES 4.0 Server. This view has been modified in 5.1 to include custom Fixlet content.

Column	Type	Description
Sitename	varchar(128)	Fixlet site name
ID	int	Fixlet ID
ComputerID	int	Computer ID
FirstBecameRelevant	datetime	Time at which Fixlet first became relevant
LastBecameRelevant	datetime	Time at which Fixlet last became relevant
LastBecameNonRelevant	datetime	Time at which Fixlet last became non-relevant
Version	int	Fixlet version

BES_RELEVANT_TASK_HISTORY

The BES_RELEVANT_TASK_HISTORY view contains an entry for every Task/computer pair (including custom Tasks) that has ever been relevant, with timestamps indicating the first time it became relevant, the last time it became relevant (the same as FirstBecameRelevant if it only became relevant once), and the last time it became non-relevant. Some of these fields may be NULL if the event in question never occurred or if it occurred before upgrading to the BES 4.0 Server.

Column	Type	Description
Sitename	varchar(128)	Source Fixlet site name
ID	int	Task ID
ComputerID	int	Computer ID
FirstBecameRelevant	datetime	Time at which Task first became relevant
LastBecameRelevant	datetime	Time at which Task last became relevant
LastBecameNonRelevant	datetime	Time at which Task last became non-relevant
Version	int	Task version

BES_FIXLET_PROPERTIES

The BES_FIXLET_PROPERTIES view lists the different properties associated with each Fixlet (including custom Fixlets), such as the severity.

Column	Type	Description
Sitename	varchar(128)	Fixlet site name
ID	int	Fixlet ID
PropertyName	varchar(32)	Property name
PropertyValue	text	Property value

Example:

- select BF.Sitename, BF.ID, BF.Name, BFP.PropertyValue AS 'Severity' from BES_FIXLETS BF, BES_FIXLET_PROPERTIES BFP where BF.SItename = BFP.Sitename AND BF.ID = BFP.ID AND BFP.PropertyName = 'Source Severity'

BES_TASK_PROPERTIES

The BES_TASK_PROPERTIES view lists the different properties associated with each Task (including custom Tasks), such as the severity.

Column	Type	Description
Sitename	varchar(128)	Source Fixlet site name
ID	int	Task ID
PropertyName	varchar(32)	Property name
PropertyValue	text	Property value

Example:

- select BT.Sitename, BT.ID, BT.Name, BTP.PropertyValue AS 'Severity' from BES_TASKS BT, BES_TASK_PROPERTIES BTP where BT.SItename = BTP.Sitename AND BT.ID = BTP.ID AND BTP.PropertyName = 'Source Severity'

Example Report Generation

The following Perl script, with the appropriate dsn name and login supplied in line #12, will access the database and print out the contents of the four principal views in HTML tables.

```
#
# Example Perl cgi script which shows the contents of a BES Database
#

use strict;
use CGI;
use DBI;
use CGI::Carp qw(fatalsToBrowser);

$| = 1;

my $dbh = DBI->connect("dbi:ODBC:bes_locke", "bigfix", "bigfix")
      or die "unable to connect to db";

print "content-type: text/html\n\n";
print "<html><body>";
print "<h1>Contents of BES Database on LOCKE</h1>";

# Print out all column headings
{
      print "<h3>Column Headings</h3>";
      print "<table width=100% bgcolor=#b0b0f0 border=1><tr>";
      print "<td>ComputerID</td><td>Name</td>";
      print "<td>Value</td><td>IsFailure</td></tr>";
      my $query  = "select ComputerID, Name, Value, IsFailure ";
         $query .= "from BES_COLUMN_HEADINGS";
      my $sth = $dbh->prepare($query);
      $sth->execute();
      my @row;
      while(@row = $sth->fetchrow_array){
            print "<tr><td>";
            print join("</td><td>", @row);
            print "</td></tr>";
      }
      print "</table>";
}

# Print out all relevant fixlets
{
      print "<h3>Relevant Fixlets</h3>";
      print "<table width=100% bgcolor=#f0b0b0 border=1>";
      print "<tr><td>Sitename</td><td>ID</td>";
      print "<td>ComputerID</td></tr>";
      my $query = "select Sitename, ID, ComputerID from BES_RELEVANT_FIXLETS";
      my $sth = $dbh->prepare($query);
      $sth->execute();
      my @row;
      while(@row = $sth->fetchrow_array){
            print "<tr><td>";
```

```
            print join("</td><td>", @row);
            print "</td></tr>";
      }
      print "</table>";
}

# Print out all actions
{
      print "<h3>Actions</h3>";
      print "<table width=100% bgcolor=#d080ff border=1>";
      print "<tr><td>ActionID</td><td>ComputerID</td>";
      print "<td>Name</td><td>Username</td><td>Start Time</td>";
      print "<td>FixletID</td><td>Sitename</td><td>ActionStatus</td></tr>";
      my $query  = "select ActionID, ComputerID, Name, Username, StartTime, ";
         $query .= "FixletID, Sitename, ActionStatus from BES_ACTIONS";
      my $sth = $dbh->prepare($query);
      $sth->execute();
      my @row;
      while(@row = $sth->fetchrow_array){
            print "<tr><td>";
            print join("</td><td>", @row);
            print "</td></tr>";
      }
      print "</table>";
}

# Print out all known fixlets
{
      print "<h3>Known Fixlets</h3>";
      print "<table width=100% bgcolor=#b0f0b0 border=1>";
      print "<tr><td>Sitename</td><td>ID</td><td>Name</td></tr>";
      my $sth = $dbh->prepare("select Sitename, ID, Name from BES_FIXLETS");
      $sth->execute();
      my @row;
      while(@row = $sth->fetchrow_array){
            print "<tr><td>";
            print join("</td><td>", @row);
            print "</td></tr>";
      }
}

print "</body></html>";
```

Index