



BigFix[®] Enterprise Suite

Database API Reference

BigFix, Inc.
Emeryville, CA

Last Modified: 8/8/07

Version 7.0

© 1998–2007 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 which 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.

Contents

CONTENTS	ii
PREFACE	1
AUDIENCE	1
ORGANIZATION OF THIS MANUAL	1
CONVENTIONS USED IN THIS MANUAL	1
VERSIONS	1
INTRODUCTION	2
VIEW SCHEMA	3
BES_FIXLETS.....	3
BES_TASKS	3
BES_ANALYSES	3
BES_BASELINES	4
BES_COMPUTERGROUPS	4
BES_COLUMN_HEADINGS.....	4
BES_RELEVANT_FIXLETS	5
BES_RELEVANT_TASKS	5
BES_RELEVANT_BASELINES.....	6
BES_ACTIONS	6
BES_RELEVANT_FIXLET_HISTORY.....	7
BES_RELEVANT_TASK_HISTORY.....	7
BES_RELEVANT_BASELINE_HISTORY	7
BES_FIXLET_PROPERTIES.....	8
BES_TASK_PROPERTIES	8
BES_BASELINE_PROPERTIES	9
BES_ACTION_DEFS	9
BES_OBJECT_DEFS	10
BES_SITEVERSIONS.....	11
WORKING WITH FIELDS XML DATA	13
WORKING WITH MANYVERSION DATA	14
EXAMPLE REPORT GENERATION	15
INDEX	18

Preface

Audience

This reference is for users of the BigFix 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 constitute 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 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. This version of the document describes the views included in the **Enterprise 1.96** database included in BES 7.0 and later.

2BINTRODUCTION

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 BigFix Console. Since these views are intended for output only, users will not 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*.

3BVIEW SCHEMA

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_ANALYSES` 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

3BVIEW SCHEMA

Column	Type	Description
Name	varchar(255)	Analysis name

Example:

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

## BES\_BASELINES

The `BES_BASELINES` view provides a list of all Baselines in the BES Database. This table is useful for joining against the `BES_RELEVANT_BASELINES` and `BES_ACTIONS` table to get the Baseline name.

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

Example:

- ```
select Sitename, ID, Name from BES_BASELINES where Sitename = 'Enterprise
Security' 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. For performance optimization, column headings with a "Value" containing more than 8000 characters will be truncated to 8000 characters in this view.

| Column | Type | Description |
|--------|------|-------------|
|--------|------|-------------|

## 3BVIEW SCHEMA

| 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 was modified in BES 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    | Varbinary(512) | The ManyVersion of this fixlet. See the section on "Working with ManyVersion Data" for details. |

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                                                                              |
| ID         | int            | Task ID                                                                                       |
| ComputerID | int            | Computer ID                                                                                   |
| Version    | Varbinary(512) | The ManyVersion of this task. See the section on "Working with ManyVersion Data" for details. |

Example:



## 3BVIEW SCHEMA

- `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\_RELEVANT\_BASELINES

The `BES_RELEVANT_BASELINES` view contains an entry for every Baseline/computer pair in which the Baseline is relevant on that computer.

| Column     | Type           | Description                                                                                       |
|------------|----------------|---------------------------------------------------------------------------------------------------|
| SiteName   | varchar(128)   | Fixlet site name                                                                                  |
| ID         | int            | Baseline ID                                                                                       |
| ComputerID | int            | Computer ID                                                                                       |
| Version    | Varbinary(512) | The ManyVersion of this baseline. See the section on "Working with ManyVersion Data" for details. |

Example:

- `select B.SiteName, B.ID, B.Name, R.ComputerID from BES_BASELINES B, BES_RELEVANT_BASELINES R where B.SiteName = R.SiteName AND B.ID = R.ID`

## BES\_ACTIONS

The `BES_ACTIONS` view contains an entry for every Action/computer pair in which the computer received the action.

| 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'`

## 3BVIEW SCHEMA

## 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 was modified in BES 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               | Varbinary(512) | The ManyVersion of this fixlet. See the section on "Working with ManyVersion Data" for details. |

## 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               | Varbinary(512) | The ManyVersion of this task. See the section on "Working with ManyVersion Data" for details. |

## BES\_RELEVANT\_BASELINE\_HISTORY

The `BES_RELEVANT_BASELINE_HISTORY` view contains an entry for every Baseline/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-

## 3BVIEW SCHEMA

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            | Baseline ID                                                                                       |
| ComputerID            | int            | Computer ID                                                                                       |
| FirstBecameRelevant   | datetime       | Time at which Baseline first became relevant                                                      |
| LastBecameRelevant    | datetime       | Time at which Baseline last became relevant                                                       |
| LastBecameNonRelevant | datetime       | Time at which Baseline last became non-relevant                                                   |
| Version               | Varbinary(512) | The ManyVersion of this baseline. See the section on "Working with ManyVersion Data" for details. |

## 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

3BVIEW SCHEMA

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'`

BES_BASELINE_PROPERTIES

The `BES_BASELINE_PROPERTIES` view lists the different properties associated with each Baseline, such as the severity.

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

Example:

- `select BB.Sitename, BB.ID, BB.Name, BBP.PropertyValue AS 'Severity' from BES_BASELINES BB, BES_BASELINE_PROPERTIES BBP where BB.Sitename = BBP.Sitename AND BB.ID = BBP.ID AND BBP.PropertyName = 'Source Severity'`

BES_ACTION_DEFS

The `BES_ACTION_DEFS` view lists all of the data associated with the latest version of each Action object in the database. The Fields column is a binary encoding of an XML document which defines the properties of the Action object. See the section on “Working with Fields XML Data” for details on how to access that data.

Column	Type	Description
ID	Int	Action ID
Version	Varbinary(512)	The ManyVersion of this action. See the section on “Working with ManyVersion Data” for details. Only the latest version of each action is contained in this view.
ParentID	Int	A value of 0 indicates that this action is a top-level action, i.e. it is not a member of a multiple action group. A value of 1 (trash) indicates that this action has been stopped. Any other value indicates that this action is a member of a multiple action group, and the value is the ID of the top-level multiple action group action for the group.
CreationTime	Datetime	The date and time that this action was taken.
Username	Varchar(32)	The SQL database username of the operator who took this action.

3BVIEW SCHEMA

Column	Type	Description
ContentType	Smallint	The value of this column should be 7 (singleActionType) or 8 (multipleActionGroupType). A value of 7 indicates that this action is a single action that may or may not be a member of a multiple action group. A value of 8 indicates that this action is the top-level action for a multiple action group. All of the member actions of the group should have a ParentID value that matches the ID of this action.
Name	Varchar(255)	The name of the action.
Fields	Image	A binary encoding of an XML document which contains all of the properties of this action. See the section on “Working with Fields XML Data” for details.
Sequence	Rowversion	The local database sequence number for this row.
OriginServerID	Tinyint	If the value is NULL, then this action was created on this DSA server, otherwise this action was replicated from another DSA server, and the value is the DSA server ID of the server on which this action was originally created.
OriginSequence	Binary(8)	If the value is NULL, then this action was created on this DSA server, otherwise this action was replicated from another DSA server, and the value is the rowversion value of the Sequence column of this action on the DSA server on which the action was originally created.

Example:

- ```
select ID, Name, dbo.fn_ExtractField('Source Fixlet',0,Fields) from
BES_ACTION_DEFS where ParentID = 0 and dbo.fn_IsActionExpired(Fields,
GetUTCDate()) = 0
```

## BES\_OBJECT\_DEFS

The BES\_OBJECT\_DEFS view lists all of the data associated with the latest version of each BES object (excluding Action objects) in the database. Note that all objects gathered from fixlet sites, as well as custom-authored objects, are included in this table. The Fields column is a binary encoding of an XML document which defines the properties of the object. See the section on “Working with Fields XML Data” for details on how to access that data. If an object has a ContentType value of 2 (fileType), then it represents a binary file. The actual binary contents of the file are stored in a separate table, not in the Fields column.

| Column   | Type         | Description                                                                                                                                                                                                                                                                                                                  |
|----------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sitename | Varchar(128) | For custom-authored objects, the value in the Sitename column is always 'ActionSite', even if the object is propagated in a different site (e.g. a custom site). For objects gathered from fixlet sites, the Sitename is the name of the fixlet site followed by '__XXX' where XXX is the version number of the fixlet site. |
| ID       | int          | Object ID                                                                                                                                                                                                                                                                                                                    |

## 3BVIEW SCHEMA

| Column         | Type           | Description                                                                                                                                                                                                                                                                                                                                     |
|----------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IsFixlet       | smallint       | True for fixlets, tasks, analyses, baselines, and computer groups.                                                                                                                                                                                                                                                                              |
| Version        | Varbinary(512) | The ManyVersion of this object. See the section on "Working with ManyVersion Data" for details. Only the latest version of each object is contained in this view.                                                                                                                                                                               |
| ParentID       | Integer        | A value of 1 (trash) indicates that the objects has been deleted.                                                                                                                                                                                                                                                                               |
| CreationTime   | Datetime       | The date and time that this version of this object was created.                                                                                                                                                                                                                                                                                 |
| Username       | Varchar(32)    | The SQL database username of the operator who created this version of this object.                                                                                                                                                                                                                                                              |
| ContentType    | smallint       | 0: site (only for object ID = 0)<br>0: fixlet type<br>2: file type<br>3: trash type (should only occur for object ID = 1)<br>5: question type (retrieved property definition)<br>6: question set type                                                                                                                                           |
| Name           | Varchar(255)   | The name of the object.                                                                                                                                                                                                                                                                                                                         |
| Fields         | image          | A binary encoding of an XML document which contains all of the properties of this object. See the section on "Working with Fields XML Data" for details.                                                                                                                                                                                        |
| Sequence       | Rowversion     | The local database sequence number for this row.                                                                                                                                                                                                                                                                                                |
| OriginServerID | tinyint        | If the value is NULL, then this version of this object was created on this DSA server, otherwise this version of this object was replicated from another DSA server, and the value is the DSA server ID of the server on which this version of this object was originally created.                                                              |
| OriginSequence | Binary(8)      | If the value is NULL, then this version of this object was created on this DSA server, otherwise this version of this object was replicated from another DSA server, and the value is the rowversion value of the Sequence column of this version of this object on the DSA server on which this version of this object was originally created. |

Example:

- ```
select Sitename, ID, Name, dbo.fn_ExtractField('Source Severity',0,Fields)
from BES_OBJECT_DEFS where ContentType = 0 and IsFixlet = 1 and NOT
ParentID = 1
```

BES_SITEVERSIONS

The `BES_SITEVERSIONS` view lists all of Fixlet sites that are currently subscribed, and the current version of those sites that have been gathered and imported into the database. Sites which have been unsubscribed are

3BVIEW SCHEMA

not included in the table, even though their contents may not yet have been removed from the object definition tables.

Column	Type	Description
Sitename	Varchar(128)	The name of the fixlet site.
Version	Int	The version of the fixlet site contents that is currently in the database. If the version is 0, then the site has been subscribed, but the contents of the site have not yet been imported into the database.

Example:

- `select * from BES_SITEVERSIONS where Version = 0`

4BWORKING WITH FIELDS XML DATA

Working with Fields XML Data

The Fields column of the BES_ACTION_DEFS and BES_OBJECT_DEFS views contains a binary encoding of an XML document that defines the properties of the object. This is essentially a collection of (name, value) pairs. There is a stored procedure called fn_ExtractField () that returns the value for a given name.

For example:

```
select dbo.fn_ExtractField('SourceSeverity', 0, O.Fields) from BES_OBJECT_DEFS O
where Sitename = 'ActionSite' and ID = 1234 and IsFixlet = 1
```

returns the value of the SourceSeverity field for the custom-authored fixlet object whose ID is 1234.

The first parameter of the function is the name of the field whose value should be returned. The second parameter is 0 if the name specifies a type of field for which there can be only one value. If the name corresponds to a type of field for which there can be multiple values, then the second parameter specifies which of the multiple values to retrieve. The third parameter is the binary-encoded XML document, which is typically a value from the Fields column.

There is another useful function called fn_IsActionExpired (). This function takes the Fields column value from an ACTION_DEF and determines if that action is expired at a given date and time.

For example:

```
select * from BES_ACTION_DEFS
where dbo.fn_IsActionExpired(Fields, GetUTCDate()) = 1
```

returns all of the actions that are expired at the current time.

Note: The function returns 1 if the action is expired, and returns 0 if the action is not expired. If no expiration time can be found in the XML document specified by the first parameter, then the function returns 0.

In addition to using these functions, you can convert the value of the Fields column to VARCHAR(8000) in order to see up to the first 8000 characters of the XML document.

For example:

```
select CONVERT(VARCHAR(8000), Fields) from BES_ACTION_DEFS
where dbo.fn_IsActionExpired(Fields, GetUTCDate()) = 1
```

returns up to the first 8000 characters of the Fields XML document for each action that is expired at the current time.

5BWORKING WITH MANYVERSION DATA

Working with ManyVersion Data

The Distributed Server Architecture, introduced in BES 7.0, uses a new version scheme in order to allow servers to modify objects independently of each other, and later resolve any conflicts that arise from those modifications by comparing the versions. A DSA version, called a ManyVersion, consists of one integer for each DSA server.

If your deployment only contains one DSA server, then all of the ManyVersion values in the database will consist of a single integer, and the Version column can be converted to an integer, and compared as an integer. If your deployment contains more than one DSA server, then objects may have ManyVersions that consist of more than one integer. The first integer is the number of times the object has been modified on the DSA server with server ID 0, the second integer is the number of times the object has been modified on the DSA server with server ID 1, and so on. This array of integers is stored in a variable length binary object (the SQL type is varbinary(512)).

There are several functions defined to help handle DSA versions:

```
fn_ManyVersionLessThan(A, B) : returns 1 if A < B, otherwise 0
fn_ManyVersionLessThanOrEqual(A, B) : returns 1 if A <= B, otherwise 0
fn_ManyVersionDominates(A, B) : returns 1 if A dominates B, otherwise 0
fn_ManyVersionConflicts(A, B) : returns 0 if A dominates B or B dominates A or A = B,
otherwise 1
```

The less than functions compare ManyVersions using a “strict” ordering such that one version will always be less than or equal to another. The “dominates” comparison is not a “strict” ordering. A dominates B if all of its version numbers are higher than or equal to B’s version numbers for every server. It is possible that A does not dominate B, while B also does not dominate A, and A is not equal to B. In this case, A and B are said to “conflict”. Conflicting versions indicate that a modification was made to an object on more than one server at “the same time” (i.e. both changes were made without knowledge of each other).

6BEXAMPLE REPORT GENERATION

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>";
    }
}
```

6BEXAMPLE REPORT GENERATION

```
        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>";
}
```

6BEXAMPLE REPORT GENERATION

```
}  
  
# 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>";
```

7BINDEX

Index

A

Action · 6, 10, 11
 Action ID · 6
 ActionStatus · 7, 17
 ADO · 2
 Analyses · 3, 4
 Analysis · 3
 Audience · 1

B

Backwards compatibility · 2
 Baseline · 4, 6, 8, 9
 BES 4.0 Server · 8
 BES Database · 1, 3–9, 2, 3, 4, 16
 BES_ACTION_DEFS · 14
 BES_ACTIONS · 3, 4, 6
 BES_ANALYSES · 3
 BES_BASELINE_PROPERTIES · 9, 10, 11, 12
 BES_BASELINES · 4
 BES_COLUMN_HEADINGS · 4, 5
 BES_COMPUTERGROUPS · 4
 BES_FIXLET_PROPERTIES · 8
 BES_FIXLETS · 3
 BES_OBJECT_DEFS · 14
 BES_RELEVANT_BASELINE_HISTORY · 8
 BES_RELEVANT_BASELINES · 4, 6
 BES_RELEVANT_FIXLET_HISTORY · 7
 BES_RELEVANT_FIXLETS · 3, 5
 BES_RELEVANT_TASK_HISTORY · 7
 BES_RELEVANT_TASKS · 3, 5
 BES_TASK_PROPERTIES · 9
 BES_TASKS · 3

C

ComputerID · 5, 6, 7, 8, 16, 17
 Conventions · 1
 custom Fixlet · 5, 7

D

Distributed Server Architecture (DSA) · 15

E

Example program · 14, 15, 16

F

Fixlet · 5, 7, 8, 12. *See*
 FixletID · 7, 17

H

HTML · 2, 16

I

Introduction · 2

M

ManyVersion Data · 6, 7, 8, 15

O

ODBC · 2, 16
 OriginServerID · 11

P

Perl · 1, 2, 16
 PropertyName · 9, 10, 12
 PropertyValue · 9, 10, 11

7BINDEX

R

relevant · 5, 6, 7, 8, 17
Report Generation · 16
Retrieved properties · 4

S

Sitename · 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 18
SQL · 2, 3

T

Task · 5, 7, 9. *See*

V

Value · 5
Version · 1, 5, 6, 7, 8, 10, 11, 12, 15
Views · 3-9, 3-9, 3-9

X

XML · 10, 11, 14
XML Data · 10, 11, 14