



# DBforBIX Setup Guide

Andrea Dalle Vacche  
Guide relative to release:  
0.4



---

## Table of Contents

1. About DBforBix.....	4
History.....	4
Author .....	4
2. Supported Databases.....	4
3. Monitoring Capabilities.....	5
on Oracle Database.....	5
on MySQL server.....	6
on PostgreSQL server.....	6
on MS SQL server .....	6
On DB2 .....	6
4. Distribution .....	7
5. Technical Details .....	7
6. Requirements.....	8
7. Generated Graphs.....	8
8. Installation .....	9
Assumptions.....	9
Common (Universal) Installation Steps.....	9
Installation on Windows .....	9
Install steps for Oracle .....	10
Install steps for PostgreSQL .....	11
Steps for Installation on MySQL.....	11
Steps for Installation on Microsoft SQL Server.....	11
Steps for Installation on IBM DB2.....	11
9. Configuration .....	12
Config.props.....	12
ZabbixServerList.....	12
DBforBIXDaemon .....	12
DatabaseList.....	13
Database Connection Parameters .....	14
Query List File.....	15



---

Configuration of query.props file.....	15
10. DBforBIX FAQ .....	17
How do I start/stop the Daemon? .....	17
How does Logging work? .....	17
Is there a way for DBforBIX to monitor RAC or DataGuard? .....	17
What can I modify without restarting the daemon? .....	17
Is it possible divide monitoring for based on database type and/or environment? .....	18



## 1. About DBforBix

DBforBIX is a daemon designed to work in combination with Zabbix Enterprise Monitor to provide multi-tiered monitoring, performance and availability reporting and measurement for the many kind of different databases, along with server performance metrics.

It provides an effective mechanism to acquire data from numerous databases installation, and in turn provides this information for monitoring and performance metrics to your Zabbix server.

You can then utilize the reporting capabilities of Zabbix for all data collected, and provide analysis such as graphs and service level agreement metrics for stakeholders.

The current distribution contains a set of pre-defined templates which incorporate alerting and graphing capabilities from initial deployment. However these can be fine tuned to suit your needs and data/monitoring requirements.

DBforBIX can run as a Windows Service and is able to work on many different environments. Where is available a JRE 1.6 you can run DbforBIX.

### History

You can read about the history and evolution of the DBforBIX plug-in in these threads below:

Original here: <http://www.zabbix.com/forum/showthread.php?t=13666>

Latest here: <http://www.zabbix.com/forum/showthread.php?t=16391>

The origins of DBforBIX can be traced though Orabbix, PostBIX, MySQLBIX and DB2Bix, and as such inherits all its predecessors benefits.

### Author

DBforBIX was developed and written by:

Andrea Dalle Vacche : <http://www.smartmarmot.com>

## 2. Supported Databases

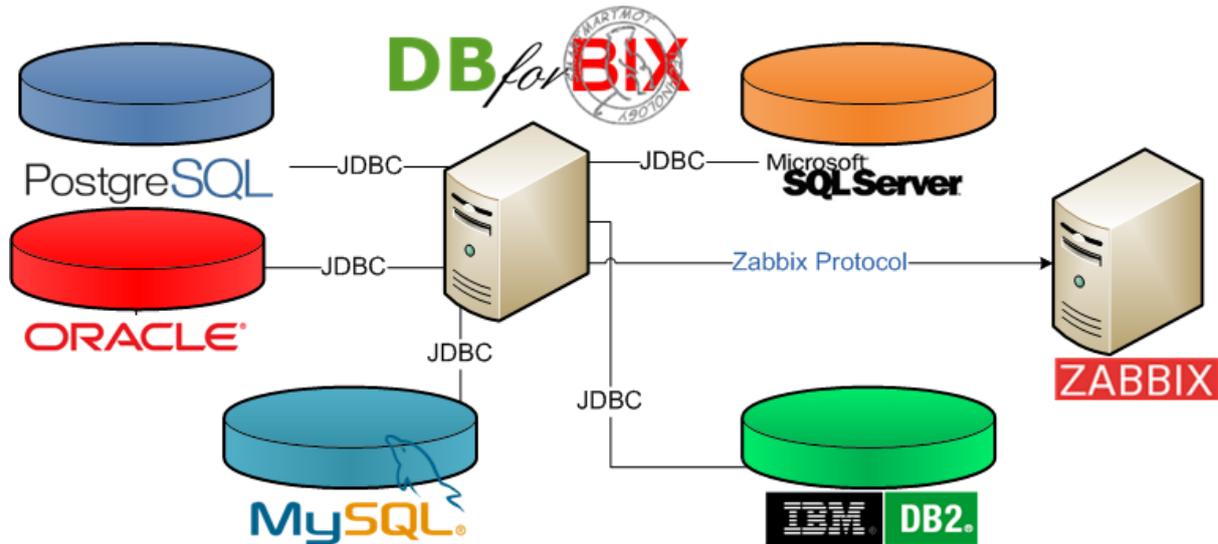
DBforBIX can acquire and retrieve every kind performance parameter and vital statistic that is supported or available through standard queries.

DBforBIX support the following databases:

- Oracle
- MySQL Server
- PostgreSQL

- MS SQL Server
- DB2

All information gathered from the monitored databases is retrieved by query, using the Java JDBC layer. This ensures compatibility and diversity with future database versions and types.



### 3. Monitoring Capabilities

Below are some examples of what DBforBIX is capable of monitoring.

#### on Oracle Database

The following are just some of parameters that DBforBIX is capable of monitoring on Oracle:

- Availability
- DB Version (for every vendor)
- Archiving (Archive log production with trend analysis)
- Event Waits (Files I/O, single block read, multi-block read, direct path read, SQLNet Messages,
- Control file I/O, Log Write)
- Hit Ratio (Hit Ratio on Triggers, Tables/Procedures, SQL Area, Body)
- Logical I/O (Server performance on Logical I/O of: Current Read, Consistent Read, Block
- Change)
- Physical I/O (Redo Writes, Datafile Writes, Datafile Reads)
- PGA
- SGA (In particular; Fixed Buffer, Java Pool, Large Pool, Log Buffer, Shared Poolm Buffer Cache)
- Shared Pool (Pool Dictionary Cache, Pool Free Memory, Library Chache, SQL Area, MISC.)

- Pin Hit Ratio (Oracle library cache pin are caused by contention with the library cache, the area
- used to store SQL executables for re-use)
- Sessions / Processes
- Sessions (Active Sessions, Inactive Sessions, System Sessions)
- DBSize/DBFileSize (DBSize size of database really used space and of Filesize)

### on MySQL server

The following are just some of parameters that DBforBIX is capable of monitoring on MySQL. The list was truncated due to the sheer volume of list items:

- Dbversion
- table/thread cache
- Com created/dropped/changed/committed
- Innodb statistics
- Key read/write/requests
- Qcache statistics

### on PostgreSQL server

The following are just some of parameters that DBforBIX is capable of monitoring on PostgreSQL:

- Availability
- Buffers (Backend, checkpoint, clean, allocated)
- Checkpoint (requested, timed)
- Connections
- Tuples (deleted, updated, fetched, returned, inserted)
- Xact (rollback, commit)
- Locks (every kind of exclusive locks and every kind of general locks)

### on MS SQL server

The following are just some of parameters that DBforBIX is capable of monitoring on Microsoft SQL Server:

- Cachehit
- I/O Pending
- waittime
- dbsize/logsize
- log usedsize
- Page reades/writes

### On DB2

Coming soon to a DBforBIX distribution near you!



## 4. Distribution

You can find the latest versions of DBforBIX locations listed below:

<http://www.smartmarmot.com/product/dbforbix/download/>

or

<https://sourceforge.net/projects/dbforbix/>

## 5. Technical Details

DBforBIX has been made to monitor and control every kind of principal database with just one daemon.

Points of force of this daemon are:

- Apache DBCP connection pool
- logging realized by Log4J
- Hyper Threading
- Superscalar (you can monitor a huge amount of databases of different kind)
- Refresh parameter while running (you don't need to restart the daemon)
- unlimited support of Zabbix servers (send the retrieved items to all Zabbix server)
- Items collision free

### Pros:

- You can customize connection pooling for each database with different parameters.
  - This means that connections are reused (constantly making a new connection introduces an overhead for all databases) and when idle or dropped.
- You can customize your logfile format as you see fit, so it can be parsed to/from your own software to maintain control of information gathered by DBforBIX logging and assist in error troubleshooting.
- Hyper Threading: If a database is slow, it won't impact your other databases as every job is a thread and all generated threads run independently.
- DBforBIX can scale on multiple processors and is really lightweight.
- You can definitively use it in your mission critical environment, since the architecture allows you to send all retrieved items to any number of Zabbix servers, without limitations.
- All the Items retrieved are collision free.
  - e.g. Alive is a true "common" item that can have collision with another Item with the same name.

Here is an example of the collision problem, solved by DBforBIX automatically introducing a prefix that is always in the form of:

"DBforBIX.<db type>.itemkey"

<dbtype> can be Oracle, MySQL, PostgreSQL, DB2, MSSQL and are predefined (you don't need to configure or modify anything)



## 6. Requirements

To use DBforBIX, you will require the following on your Zabbix Server:

- Zabbix 1.8.x Server
- Java Runtime Environment 6
- DBforBIX distribution binaries

Current known successfully tested DBforBIX host platforms:

- RHEL5.X (CentOS 5.4 & 5.5)
- Windows 2003 (with Java SE 1.6)
- HP-UX 11.31
- AIX 5.3

Currently known tested Oracle Versions:

- 9i
- 10g
- 10.2
- 11G

Currently known tested PostgreSQL Versions:

- 8.3
- 8.4
- 9.0

Currently known tested MySQL Versions:

- 5.x

Currently known tested SQLServer Versions:

- 2005
- 2000

### **\*\* IMPORTANT \*\***

You do NOT need to install any client of any kind for DBforBIX to work.

You need to download and place under /lib/ directory the following library:

- db2jcc\_license\_cu.jar
- db2jcc.jar
- ojdbc6.jar

DBforBIX uses Java objects and connection strings to connect to the Oracle Database, and as such doesn't require any database client to be installed on your Zabbix Server.

## 7. Generated Graphs

There are a significant number of graphs generated by the default templates included in DBforBIX, for each kind of database.

Some examples are coming soon.

## 8. Installation

### Assumptions

The installation instructions have been created based on instructions for most \*nix deployments (i.e. RHEL/CentOS), and is assumed that any alterations required for your specific environment are to be taken into account accordingly.

This guide is based upon there being two (2) Hosts, 1x Zabbix Server and 1x database Server. If you are planning on monitoring a database instance that is running on your Zabbix Server, the steps are the same, with minor adjustments required for your connection information.

The steps also assume you are configuring DBforBIX to monitor a new installation or setup of Oracle. The installation steps will have you GRANT access for the Zabbix user to all tables, and this will include any USER tables present at the time of execution.

If you do not want Zabbix to have access to specific tables or resources within your database, you will need to set a DENY to the Zabbix users access as required. Consult your DBA for details, as this is beyond the scope of these instructions.

### Common (Universal) Installation Steps

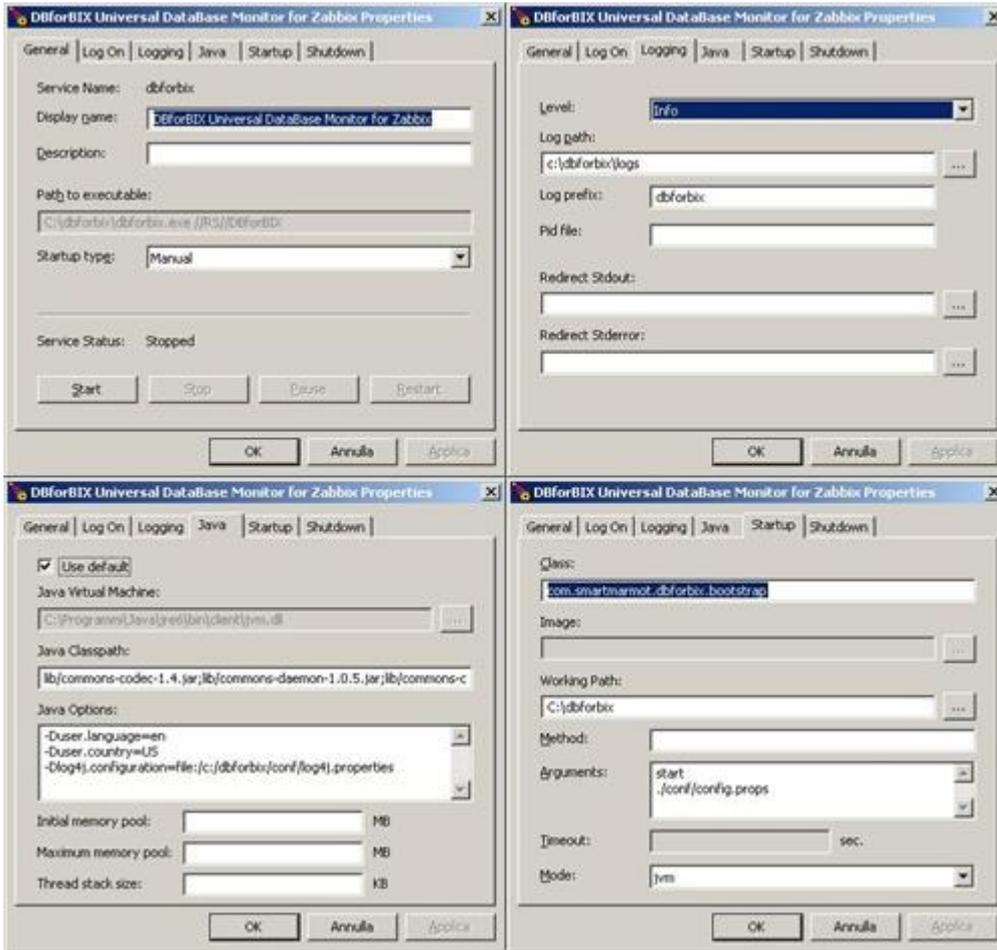
- Download DBforBIX to your Zabbix Server
- On your Zabbix server, unzip DBforBIX to: /opt/dbforbix
- Copy file /opt/dbforbix/init.d/dbforbix to /etc/init.d/dbforbix
- Grant execute permissions to the following files:
  - /etc/init.d/dbforbix
  - /opt/dbforbix/run.sh
    - For this example on RedHat, run:  
**chkconfig -add dbforbix**
    - Verify with:  
**chkconfig -list**
- Create a Host entry in Zabbix for the DB Instance/s you are planning to monitor, and import your needed templates found at: /opt/dbforbix/template

### Installation on Windows

DBforBIX implements an Apache daemon and offers the capability to run as a service on Microsoft Windows environments.

Install DBforBIX on Windows is really easy just follow these steps:

- Create a directory on c: called dbforbix
- extract the distribution inside "c:\dbforbix"
- double click on "install.cmd"
- after you'll find a service called "DBforBIX Universal Database Monitor for Zabbix"



You can use

“dbforbixw.exe” to customize service's parameters if required. Refer Figures 1.a for screenshots that show dbforbixw.exe at work.

Figure 1.a

### Install steps for Oracle

Create a User (ZABBIX) for DBforBIX to access your Oracle Database. You can use the following script:

```

CREATE USER ZABBIX
  IDENTIFIED BY <REPLACE WITH PASSWORD>
  DEFAULT TABLESPACE SYSTEM
  TEMPORARY TABLESPACE TEMP
  PROFILE DEFAULT
  ACCOUNT UNLOCK;
-- 2 Roles for ZABBIX
GRANT CONNECT TO ZABBIX;
GRANT RESOURCE TO ZABBIX;
  
```

```
ALTER USER ZABBIX DEFAULT ROLE ALL;
- 5 System Privileges for ZABBIX
GRANT SELECT ANY TABLE TO ZABBIX;
GRANT CREATE SESSION TO ZABBIX;
GRANT SELECT ANY DICTIONARY TO ZABBIX;
GRANT UNLIMITED TABLESPACE TO ZABBIX;
GRANT SELECT ANY DICTIONARY TO ZABBIX;
```

**\*\* NOTE :** If you are using Oracle 11g, you will need to add the following:

```
exec dbms_network_acl_admin.create_acl(acl => 'resolve.xml',description => 'resolve acl', principal
=>'ZABBIX',is_grant => true, privilege => 'resolve');
exec dbms_network_acl_admin.assign_acl(acl => 'resolve.xml', host => '*');
```

You can verify the above is correct by running:

```
select utl_inaddr.get_host_name('127.0.0.1') from dual;
```

## Install steps for PostgreSQL

Create a User (ZABBIX) for DBforBIX to access your PostgreSQL Database. You can use the following script:

```
CREATE USER zabbix WITH PASSWORD 'passw0rd';
GRANT SELECT ON pg_stat_activity to zabbix;
GRANT SELECT ON pg_stat_activity to zabbix;
GRANT SELECT ON pg_database to zabbix;
GRANT SELECT ON pg_authid to zabbix;
GRANT SELECT ON pg_stat_bgwriter to zabbix;
GRANT SELECT ON pg_locks to zabbix;
GRANT SELECT ON pg_stat_database to zabbix;
```

## Steps for Installation on MySQL

```
CREATE USER 'zabbix_monitor'@'%'.mydomain.com IDENTIFIED BY 'zabbixpassword';
GRANT SELECT, SHOW VIEW ON *.* TO 'zabbix_monitor'@'%'.mydomain.com;
```

## Steps for Installation on Microsoft SQL Server

```
CREATE LOGIN <login name> WITH PASSWORD = '<password>' ; GO
GRANT GRANT VIEW SERVER STATE TO <login name>;GO
GRANT CONNECT TO <login name>;GO
```

## Steps for Installation on IBM DB2

Coming soon

## 9. Configuration

Now we need to configure your DBforBIX setup.

The tags below are listed as they will appear within the respective configuration files. The first you will need to modify is your config.props file to define your connection properties for Zabbix and your databases.

The config.props file can be found at `/opt/dbforbix/conf/config.props`

### Config.props

The file which is included in DBforBIX distribution if is the first installation need be renamed removing “.sample”.

With this way you will never find out overwritten your configurations file updating DBforBIX with a new release.

#### ZabbixServerList

##### #comma separed list of Zabbix servers

The settings under this tag allow you to configure your Zabbix Server information. DBforBIX can also be configured to send your Oracle data to multiple Zabbix servers.

**TIP:** This can be beneficial for distributed monitoring scenarios, server migrations or replicating DBforBIX data to a Disaster Recovery site or server.

Replace with your Zabbix Server info where appropriate;

**ZabbixServerList=ZabbixServer1,ZabbixServer2**

**ZabbixServer1.Address=192.168.0.1**

**ZabbixServer1.Port=10051**

**ZabbixServer2.Address=192.168.0.2**

**ZabbixServer2.Port=10051**

#### DBforBIXDaemon

Entries under this tag allow you to set your DBforBIX Daemon parameters.

##### #MaxThreadNumber should be >= than the number of your databases

Set the number of threads the DBforBIX Daemon should have inside his internal pool of DB Jobs. This number *should* be at least equal to (or more than) then number of databases monitored by DB, now this parameter

*if not set is automatically calculated.*

e.g. For 50 Databases, we are using 100 threads

**DBforBIX.MaxThreadNumber=100**

##### #pidFile

Next you can set the location of the Daemons PID file. Default location is:  
`/opt/dbforbix/logs`

---

**DBforBIX.PidFile=./logs/dbforbix.pid**

**\*\* NOTE:**

Relative path is permitted here

**TIP:** With the features of the DBforBIX Daemon, it is possible to clone DBforBIX and use different configurations. As such, it is therefore possible to have one Daemon that checks your TEST databases, another iteration that checks your DEVELOPMENT databases and a third that checks your PRODUCTION databases, each with different timings and check loops!

### DatabaseList

Entries under this heading allow you to configure your database list information:

**#put here your databases in a comma separated list**

This is where you define your database instances. You can specify more than one instance here, separated using a comma.

**DatabaseList=EXAMPLE1,...**

**\*\* NOTE:**

The names of the instances must match those you have specified as your HOST name in Zabbix

**#Configuration of Connection pool**

From here, you will configure settings that are specific to the connection pool. As the comments in config.props suggest, if you do not specify these values, DBforBIX will use default values which have been hard-coded.

**#Maximum number of active connection inside pool**

Set the maximum number of connections that can be allocated to this pool at any time, or alternatively set a negative value for no limit.

**DatabaseList.MaxActive=10**

**#The maximum number of milliseconds**

Here you define how long that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception, or set the value  $\leq 0$  to wait indefinitely.

**DatabaseList.MaxWait=100**

Also under the same section, you can define the maximum number of connections that can remain idle within the connection pool, without being released.

Alternatively, you can set a negative value for no limit.

**DatabaseList.MaxIdle=1**

**TIP:** You can specify the Database connection parameters for each database you wish to monitor individually, underneath your Database Connection Parameters. This allows you to customize your connection settings based on the Database constraints I.e one connection pool for your PRODUCTION DB and another for your TEST DB, for example;

`DB1.MaxActive=10`

`DB1.MaxWait=100`

`DB1.MaxIdle=1`

Note that any settings defines this way will override the general settings for your generic connection pool.

### Database Connection Parameters

This section sets your connection string to the Oracle Database. This string invokes a Java Database Connector (JDBC) to your Oracle Databases, and as such does not need the Oracle Client to be installed.

#### #define here your connection string for each database

Here you will define the connection string. These are formatted as: `DBName.Url`(as specified in your `DatabaseList`) followed by the `jdbc` string and your Database Server information .

**EXAMPLE1.Url**=write here your database Url should be in the form:

Oracle = `jdbc:oracle:thin:@<host>:<LISTENER_PORT>:<instance>`

PostgreSQL = `jdbc:postgresql://<host>:<port>/<database>`

MS Sql Server = `jdbc:jtds:sqlserver://<host>:<port>/<instancename>`

MySQL Server = `jdbc:mysql://[host:port],[host:port].../[database]`

DB2 = `jdbc:db2://<servername>:<port>/<installation>`

Set your Database username and password below. e.g.

**EXAMPLE1.User**=`zabbix`

**EXAMPLE1.Password**=`zabbix_password`

**EXAMPLE1.DatabaseType**=`<databasetype>`

where `<databasetype>` can be = `[oracle][pgsql][mssql][mysql][db2]`

it depend from your database

**EXAMPLE1.QueryListFile**=`<path to query file>`

**TIP:** Setting connections to multiple databases is made easy by adding more connection strings, with their corresponding credentials. Example given below.

#### \*\* Note:

After these entries, you can set your MaxActive, MaxWait and MaxIdle for the individual database connections if you so wish.

## Query List File

This defines where the file containing the databases specific queries (SQL) queries can be found. This is a customizable file.

You need to specify the query parameter file:

**<DBNAME>.QueryListFile=./confQueryTest.props**

*This entry will specify the query file of a database relative path are allowed.*

**TIP:** *This is really useful to use different query file for different databases or to have a pool of query file one for each Database Release (if you have different release) and/or to have different query files for Production environment, Test environment, Development environment etc..*

**\*\* NOTE:**

Most common query files are included in DBforBIX distribution you need to rename and remove ".sample"

## Configuration of query.props file

The query.props file can be modified or added to, so you can supply your own customized queries through DBforBIX against your Oracle instances/databases. Each query created has an associated 'Item' or item name that Zabbix will use to identify the query.

**\*\* NOTE:**

Item names must be unique. The configurable items are formatted as follows.

### DefaultQueryPeriod

You have to set the default query period. This value modify the period for all the query contained in query.props file.

### QueryList

You have to set the query name under the QueryList in the query.props file. Each query name is comma separated. For example;

**QueryList=queryName1,queryName2,queryName3**

### Query

You must identify the query by the unique item name you specified in the QueryList, followed by '.Query' for DBforBIX to recognise that this is the query string. It's important to remember NOT to add the semi-colon ";" to the end of your custom query.

**customQueryItemName.Query=yourQueryHere**

Now you can define what you want DBforBIX to return to your Zabbix Server if no data is found for your query.

**customQueryItemName.NoDataFound=none**

**TIP:** *In the example above, DBforBIX would send the string "none" to the Zabbix Server.*

You can specify if you want a different execution period for your query  
**customQueryItemName.Period=<Express a period in minute>**

**TIP:** You can change the execution time of a database type that may not change or update all that often, and reduce traffic/resource requirements.

Next you can specify a query that will be executed and if return 'RaceConditionValue' the query 'customQueryItemName' is executed, otherwise it is skipped

**customQueryItemName.ACTIVE=[true|false]**

if true query is executed otherwise skipped

```
archive.Query=select round( A.LOGS*B.AVG/1024/1024/10 )
from ( SELECT COUNT (*) LOGS FROM v$log_history WHERE
FIRST_TIME >= (sysdate -10/60/24)) A,
( SELECT Avg(BYTES) AVG, Count(1), Max(BYTES)
Max_Bytes,Min(BYTES) Min_Bytes
FROM v$log) B
```

```
archive.RaceConditionQuery=select value
from
v$parameter where name='log_archive_start'
archive.RaceConditionValue=FALSE
```

**TIP:** In the example above, DBforBIX would execute "archive.Query" only if the query "archive.RaceConditionQuery" returns the value "archive.RaceConditionValue". The "Archive" query will only execute if the database is in archivelogmode (Parameter "log\_archive\_start" set to TRUE)

**<QueryName>.Trim=[true|false]**

if true the resultset is trimmed (default is true)

**<QueryName>.AddSpaces=[true|false]**

if true then add a space between columns of resultset(default is true)

**<QueryName>.ExcludeColumnsList=[1,2,3,..n]**

exclude from result set the 1st, the 2nd etc.. columns from resultset

## 10. DBforBIX FAQ

### How do I start/stop the Daemon?

To start the DBforBIX Daemon, run:  
**/etc/init.d/dbforbix start**

To stop the DBforBIX Daemon, run:  
**/etc/init.d/dbforbix stop**

### How does Logging work?

The DBforBIX daemon outputs its log file to the default location of  
**/opt/dbforbix/logs/**

Logging properties can be modified by making your required changes to;  
**/opt/dbforbix/conf/log4j.properties**

The property setting responsible for defining the output location is;  
log4j.appender.DBforBIX.File=logs/dbforbix.log

**TIP:** From this properties file, you can modify the location, file name and log format as desired. For additional information, please refer to the official log4j documentation, found at;  
<http://logging.apache.org/log4j/1.2/index.html>

### Is there a way for DBforBIX to monitor RAC or DataGuard?

Yes is possible, for example;

If you have two hosts, RAC1 and RAC2, in one instance of RACINST you should write the connection string as follow:

```
RACINST.Url=jdbcrcacle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=RAC1.EXAMPLE.COM)(PORT=1521)))(ADDRESS=(PROTOCOL=TCP)(HOST=RAC2.EXAMPLE.COM)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=RACINST)))
```

### What can I modify without restarting the daemon?

The parameters that are dynamically read at each iteration of "DBforBIXDaemon.Sleep" are as follows;

- Any Query added to query.props
- Modifications to config.props

*Available Upcoming Release; To be advised*

- Database List

Currently, the only known items that don't dynamically update are the *ZabbixDaemon.MaxThreadNumber*, and changes to Connection Pool info.



---

## Is it possible divide monitoring for based on database type and/or environment?

Yes it's possible and it's easy! Basically, you just need to copy your installation directory into a new one

e.g. you can have

- /opt/dbforbix\_prod
- /opt/dbforbix\_test
- /opt/dbforbix\_devel

and then you need to copy /etc/init.d/dbforbix into

- /etc/init.d/dbforbix\_prod
- /etc/init.d/dbforbix\_test
- /etc/init.d/dbforbix\_devel

After this, you need to customize start/stop script to locate the right directory for each instance of DBforBIX

e.g.

dbforbix=/opt/dbforbix

should be changed into:

- /opt/dbforbix\_prod
- /opt/dbforbix\_test
- /opt/dbforbix\_devel

Now you have completely divided your monitoring solution to the most common scenarios of:

- development environment
- test/quality/pre-production environment
- production environment

After that you can customize your queries file for you different environments, ensure each fulfils the requirements based on their varying needs. You can play around with the suggestions above and do the same as needed.

Document Copyright © Andrea Dalle Vacche and Jason Chatfield

## License Information

DBforBIX is released under and according to the  
**GNU GENERAL PUBLIC LICENSE**  
**Version 3, 29 June 2007**