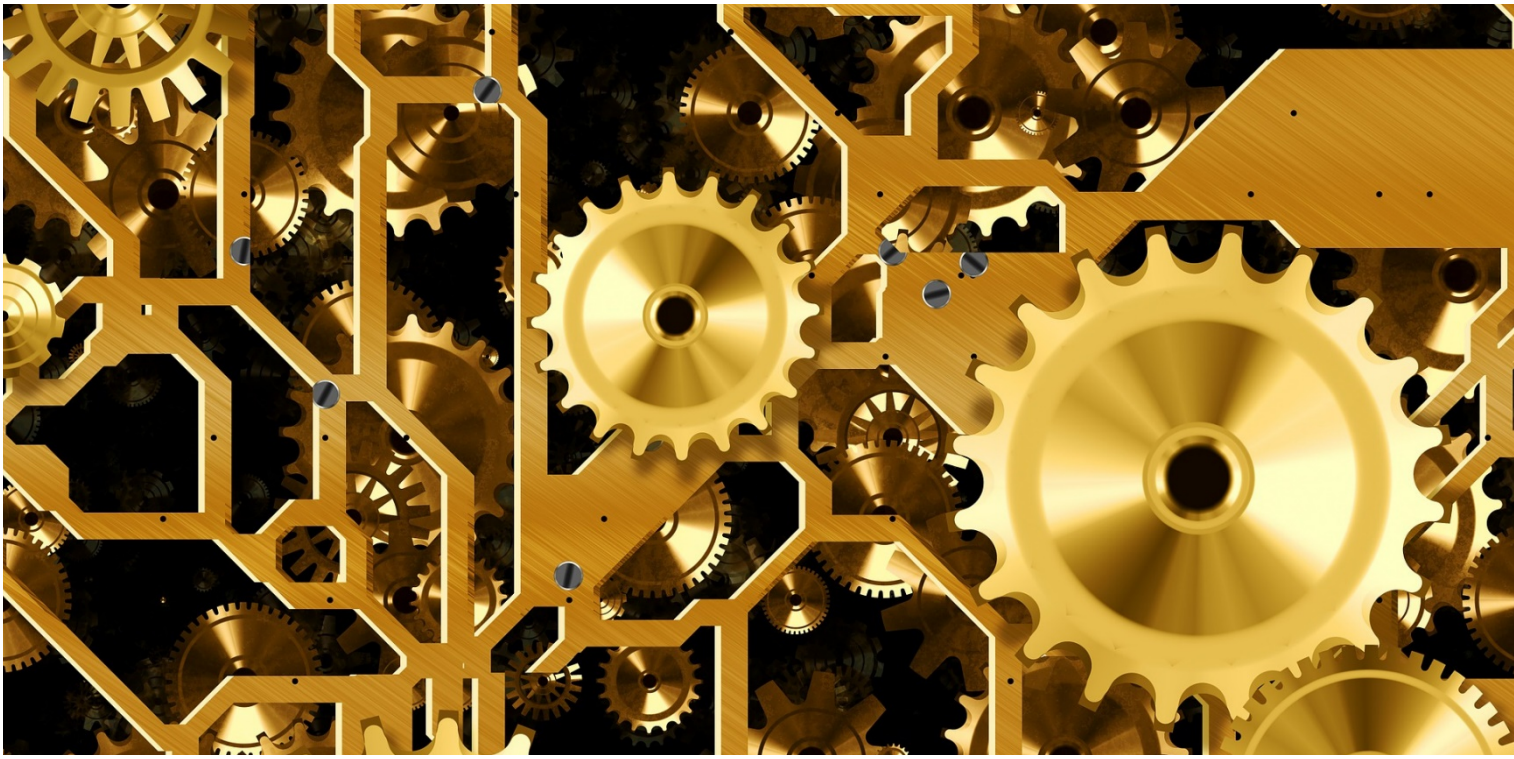

BigFix Maintenance Guide

an HCL product

Document version 9.x.2

Mark Leitch
BigFix Performance & Secure Engineering

PRODUCTS & PLATFORMS
by HCL Technologies

HCL

This edition applies to version 9.2 of BigFix and all subsequent releases and modifications until otherwise indicated.

CONTENTS

LIST OF FIGURES

REVISION HISTORY

Date	Version	Revised By	Comments
December 1st, 2018	9.x.1	MDL	Initial distribution.
January 8th, 2020	9.x.2	MDL	Version updates.

Figure 1: Revision History

1 Introduction

Maintenance is important for all software products. Systems tend towards entropy, and while many aspects of BigFix are self-managing, some core maintenance principles will ensure an installation will run at peak efficiency and capability.

We will describe some maintenance approaches for BigFix with a primary focus on the database management system. BigFix has offerings on both Linux and Windows, and specific recommendations will be given for each.

Note: The following documents are considered to offer a suite of performance, capacity, and maintenance reference information for BigFix. See the References section for relevant URLs.

- BigFix Maintenance Guide (this document).
  A set of configuration and maintenance recommendations for BigFix.
- BigFix Capacity Planning.
  A set of capacity planning and configuration recommendations for BigFix.
- MX Performance Toolkit for BigFix.
  A set of tools and performance management approaches for BigFix.

Note: While maintenance recommendations are intended to be general purpose, this document is best oriented towards BigFix Version 9.5 onwards. In addition, a number of references are provided in the References section. These papers are highly recommended for readers who want detailed knowledge of BigFix server configuration, architecture, and capacity planning.

2 Supported Database Versions

BigFix has a well-documented matrix of system requirements, including operating system versions, hypervisor versions, browser versions, and database versions. In terms of database versions, the following offerings are supported.

- Microsoft SQL Server 2008, 2012, 2014, 2016.
- DB2 Workgroup Edition 10.5.

The versions supported for a specific BigFix release are documented in the system requirements matrix (URL). The general recommendation is to use the most current database release supported for your BigFix version, as database performance, resilience, and function tend to only improve with each new release.

It should be noted that Microsoft SQL Server Express is also included in the list of reference databases. Both DB2 and Microsoft SQL Server offer “express versions”. These are license free, limited utility offerings typically intended for low demand or proof of concept situations.

In the case of DB2 Express, there is no compelling reason to use it. It is simply a constrained version of DB2, and the license for the full version is provided for BigFix Linux deployments.

In the case of Microsoft SQL Server Express, the support matrix clearly indicates it may be used for evaluation purposes, and the customer will provide the full Microsoft SQL Server license. What does this mean in the context of a BigFix deployment? Essentially, Microsoft SQL Server Express may be used for a BigFix deployment with the following constraints.

- The user must be aware of the Microsoft SQL Server Express constraints. The constraints for a specific version are documented in the Microsoft Knowledge Base (e.g. URL). In general, the DBMS is constrained to a single CPU socket and up to four cores, utilizing up to 1GB of RAM and 10GB of database storage. Once the Microsoft SQL Server Express limits are reached, the configuration is no longer supported by BigFix.
- In terms of the scale limits for BigFix with Microsoft SQL Server Express, scale on the order of 100 devices with one or two operators is expected. Even at this level of “proof of concept” scale, system monitoring is critical to ensure system health. For example, it may be possible to exceed 100 devices with careful monitoring, but it is considered a good rule of thumb. In addition, some BigFix function such as the BigFix WebUI should be enabled with care. Further detail on managing monitoring and the WebUI is provided in the following points.
- The user must perform adequate system monitoring to ensure the database system limits are not impacting the health of BigFix. For example, once the 10GB storage limit is reached, the database will no longer be viable. When the CPU and memory limits are reached, system response times and throughput will degrade. As a result system monitoring is critical. To monitor the storage limits, the Windows file explorer may be used. To manage the CPU and memory limits, the Windows performance monitor or task manager may be used for the SQL Server process. For advanced users, the Microsoft SQL Server Management Studio may be used for monitoring.
- In the event it is desired to enable the BigFix WebUI functionality, it should be done with caution. The reason for this is the WebUI will initiate additional database workload that will increase system resource requirements. Before enabling the BigFix WebUI, the base workload should be running with some headroom with respect to the Microsoft SQL Server Express limitations (CPU, memory, and storage). If resource utilization issues are indicated, either pre or post WebUI enablement, it is recommended to upgrade to the licensed version of Microsoft SQL Server.
- In the event the defined limits are reached, Microsoft does support an in place upgrade approach. As a result, a maintenance outage may be taken, the DBMS licensed, and service may be resumed.

3 Database Maintenance Approaches

There is a fair bit of commonality in maintenance approaches for different database implementations, especially in the SQL database “family”. For example. the following figure offers the general maintenance categories, and the comparable DB2 and Microsoft SQL Server utilities involved.

Maintenance Approach	DB2 Utility	Microsoft SQL Server Utility
Backup Management	backup, restore	backup, restore
Statistics Management	runstats	create or update statistics
Reorganization	reorg	alter index with the reorganize or rebuild option
Maintenance Automation	Based upon operating system and database schedulers.	

Figure 2: Database Maintenance Approaches

Some general best practices follow.

1. Define a maintenance period for the database. While online maintenance utilities are now common, they sometimes require “upgrade” editions, or still require exclusive locks to complete their operations. A weekly maintenance interval can ensure trouble-free long-term operation.
2. Automate the operations, either through the database or operating system level schedulers.
3. Periodically verify the automation work as expected. Ideally, have it notify on failure. Even then it is possible for the notification service to fail.

4 MS SQL Maintenance

The MS SQL examples use Transact-SQL syntax. However, the SQL Server Management Studio provides a graphical interface for management that may be preferred.

4.1 MS SQL Database Backup Management

It is recommended that nightly database backups be taken. The following figures offer a sample database offline backup (utilizing compression), along with a sample restore.

```
backup database <dbname> to disk='<backup path>' with compression;
```

Figure 3: MS SQL Database Backup with Compression Command

```
restore database <dbname> from disk='<backup path>';
```

Figure 4: MS SQL Database Backup Restore

4.2 MS SQL Statistics Management

MS SQL offers a comprehensive stored procedure to force statistics updates.

```
exec sp_updatestats;
```

Figure 5: MS SQL Database Statistics Collection Command

4.3 MS SQL Database Reorganization

BigFix provides an intelligent script that will either reorganize or rebuild the indexes, depending on their fragmentation. For example:

- Do nothing if the fragmentation is below 5%.
- Reorganize the index if fragmentation is between 5% and 30%.
- Rebuild the index if fragmentation exceeds 30%.

The script should be visible under the “SQL Server Agent” and is named “BFEnterprise Full Database Index Reorganization”. If the script is not in evidence, it is recommended to contact support.

- Prior to BigFix 9.5.10, the script does an offline operation.
- With BigFix 9.5.10, a number of improvements have been made to the script.
    - The database entitlement is checked and if online operations are possible, they are used.
    - The script will not alter indexes below certain page thresholds. This avoids churn and repetitive, useless maintenance operations.
    - The script has improved error handling for failed operations.
    - The script has an improved locking model, and will back off in the event locks may not be acquired (versus attempting to acquire locks indefinitely and impacting the production workload).

5 DB2 Maintenance

5.1 DB2 Database Backup Management

It is recommended that nightly database backups be taken. The following figures offer a sample database offline backup (utilizing compression), along with a sample restore.

```
backup db <dbname> user <user> using <password> to <backup directory> compress
```

Figure 6: DB2 Database Backup with Compression Command

```
restore db <dbname> from <backup directory> taken at <timestamp> without
prompting
```

Figure 7: DB2 Database Offline Backup Restore

5.2 DB2 Online Backup Support

Online backups are typically desired for full application availability. Prior to the release of BigFix 9.5.3, not all database data types were logged (to be specific, Large Object types, or LOBs, were not logged). The logging of all necessary data types is required to ensure the integrity of online backups, given they depend on the log content.

However, with the BigFix 9.5.3 release, all necessary data types are logged for new installations. As a result, these installations may safely enable online backups. On the other hand, legacy installations, even if they upgrade to BigFix 9.5.3, will typically not log all data types. As a result, online backups are not recommended for them.

For these legacy installations, Appendix A contains a set of prescribed steps to support the enablement of online backups. These steps include how to determine if specific data types are not logged, as well as a database migration procedure to enable logging for specific columns. Once the migration is complete, online backups may also be enabled for legacy installations.

5.2.1 Performing Online Backups

In the event online backups are enabled, the following figure provides commands that comprise a sample weekly schedule. With the given schedule, the best-case scenario is a restore requiring one image to restore (Monday failure using the Sunday night backup). The worst-case scenario would require four images (Sunday + Wednesday + Thursday + Friday). An alternate approach would be to utilize a full incremental backup each night to make the worst-case scenario two images. The trade-offs for the backup approaches are the time to take the backup, the amount of disk space consumed, and the restore dependencies. A best practice can be to start with nightly full online backups, and introduce incremental backups if time becomes an issue.

```
(Sun) backup db <dbname> online include logs use tsm
(Mon) backup db <dbname> online incremental delta use tsm
(Tue) backup db <dbname> online incremental delta use tsm
(Wed) backup db <dbname> online incremental use tsm
(Thu) backup db <dbname> online incremental delta use tsm
(Fri) backup db <dbname> online incremental delta use tsm
(Sat) backup db <dbname> online incremental use tsm
```

Figure 8: DB2 Database Online Backup Schedule

Note to enable incremental backups, the database configuration must be updated to track page modifications, and a full backup taken in order to establish a baseline.

```
update db cfg for BFENT using TRACKMOD YES
```

Figure 9: DB2 Database Incremental Backup Enablement

To restore the online backups, either a manual or automatic approach may be used. For the manual approach, you must start with the target image, and then revert to the oldest relevant backup and move forward to finish with the target image. A far simpler approach is to use the automatic option and let DB2 manage the images. A sample of each approach is provided below, showing the restore based on the Thursday backup.

```
restore db <dbname> incremental use tsm taken at <Sunday full timestamp>
restore db <dbname> incremental use tsm taken at <Wednesday incremental
timestamp>
restore db <dbname> incremental use tsm taken at <Thursday incremental delta
timestamp>
```

Figure 10: DB2 Database Online Backup Manual Restore

```
restore db <dbname> incremental auto use tsm taken at <Thursday incremental delta
timestamp>
```

Figure 11: DB2 Database Online Backup Automatic Restore

In order to support online backups, archive logging must be enabled. The next subsection provides information on archive logging, including the capability to restore to a specific point in time using a combination of database backups and archive logs.

5.3 Database Log Archiving

A basic approach we will advocate is archive logging with the capability to support online backups. The online backups themselves may be full, incremental (based on the last full backup), and incremental delta (based on the last incremental backup). In order to enable log archiving to a location on disk, the following command may be used.

```
update db cfg for <dbname> using logarchmeth1 DISK:/path/logarchive
```

Figure 12: DB2 Database Log Archiving to Disk

Alternatively, in order to enable log archiving to TSM, the following command may be used[1].

```
update db cfg for <dbname> using logarchmeth1 TSM
```

Figure 13: DB2 Database Log Archiving to TSM

Note that a “logarchmeth2” configuration parameter also exists. If both of the log archive method parameters are set, each log file is archived twice (once per log archive method configuration setting). This will result in two copies of archived log files in two distinct locations (a useful feature based on the resiliency and availability of each archive location).

Once the online backups and log archive(s) are in effect, the recovery of the database may be performed via a database restore followed by a roll forward through the logs. Several restore options have been previously described. Once the restore has been completed, roll forward recovery must be performed. The following are sample roll forward operations.

[1] The log archive methods (logarchmeth1, logarchmeth2) have the ability to associate configuration options with them (logarchopt1, logarchopt2) for further customization.

```
rollforward <dbname> to end of logs
```

Figure 14: DB2 Database Roll Forward Recovery: Sample A

```
rollforward <dbname> to 2012-02-23-14.21.56 and stop
```

Figure 15: DB2 Database Roll Forward Recovery: Sample B

It is worth noting the second example recovers to a specific point in time. For a comprehensive description of the DB2 log archiving options, the DB2 information center should be consulted (URL). A service window (i.e. stop the application) is typically required to enable log archiving.

5.4 DB2 Database Backup Cleanup

Unless specifically pruned, database backups may accumulate and cause issues with disk utilization or, potentially, a stream of failed backups. If unmonitored backups begin to fail, it may make disaster recovery near impossible in the event of a hardware or disk failure. A simple manual method to prune backups follows.

```
find /backup/DB2 -mtime +7 | xargs rm
```

Figure 16: DB2 Database Backup Cleanup Command

A superior approach is to let DB2 automatically prune the backup history and delete your old backup images and log files. A sample configuration is provided below.

```
update db cfg for BFENT using AUTO_DEL_REC_OBJ ON
update db cfg for BFENT using NUM_DB_BACKUPS 21
update db cfg for BFENT using REC_HIS_RETENTN 180
```

Figure 17: DB2 Database Backup Automatic Cleanup Configuration

It is also generally recommended to have the backup storage independent from the database itself. This provides a level of isolation in the event volume issues arise (e.g. it ensures that a backup operation will not fill the volume hosting the tablespace containers, which could possibly lead to application failures).

5.5 DB2 Database Statistics Management

As discussed in the previous performance management section, database statistics ensure that the DBMS optimizer makes wise choices for database access plans. The DBMS is typically configured for automatic statistics management. However, it may often be wise to force statistics as part of a nightly or weekly database maintenance operation. A simple command to update statistics for all tables in a database is the “reorgchk” command.

```
reorgchk update statistics on table all
```

Figure 18: DB2 Database Statistics Collection Command

One issue with the reorgchk command is it does not enable full control over statistics capturing options. For this reason, it may be beneficial to perform statistics updates on a table by table level. However, this can be a daunting task for a database with hundreds of tables. As a result, the following SQL statement may be used to generate administration commands on a table by table basis.

```
select 'runstats on table ' || STRIP(tabschema) || '.' || tabname || ' with
distribution and detailed indexes all;' from SYSCAT.TABLES where tabschema in
('DBO');
```

Figure 19: DB2 Database Statistics Collection Table Iterator

5.6 DB2 Database Reorganization

Over time, the space associated with database tables and indexes may become fragmented. Reorganizing the table and indexes may reclaim space and lead to more efficient space utilization and query performance. In order to achieve this, the table reorganization command may be used. Note, as discussed in the previous performance management section, automatic database reorganization may be enabled to reduce the requirement for manual maintenance.

The following commands are examples of running a “reorg” on a specific table and its associated indexes. Note the “reorgchk” command previously demonstrated will actually have a per table indicator of what tables require a reorg. Using the result of “reorgchk” per table reorganization may be achieved for optimal database space management and usage.

```
reorg table <table name> allow no access
reorg indexes all for table <table name> allow no access
```

Figure 20: DB2 Database Reorganization Commands

It is important to note there are many options and philosophies for doing database reorganization. Every enterprise must establish its own policies based on usage, space considerations, performance, etc. The above example is an offline reorg. However, it is possible to also do an online reorg via the “allow read access” or “allow write access” options. The “notruncate” option may also be specified (indicating the table will not be truncated in order to free space). The “notruncate” option permits more relaxed locking and greater concurrency (which may be desirable if the space usage is small or will soon be reclaimed). If full online access during a reorg is required, the “allow write access” and “notruncate” options are both recommended.

Note it is also possible to use our table iteration approach to do massive reorgs across hundreds of tables as shown in the following figure. The DB2 provided snapshot routines and views (e.g. SNAPDB, SNAP_GET_TAB_REORG) may be used to monitor the status of reorg operations.

```
select 'reorg table ' || STRIP(tabschema) || '.' || tabname || ' allow no
access;' from SYSCAT.TABLES where tabschema in ('DBO');

select 'reorg indexes all for table ' || STRIP(tabschema) || '.' || tabname || '
allow no access;' from SYSCAT.TABLES where tabschema in ('DBO');
```

Figure 21: DB2 Database Reorganization Table Iterator

APPENDIX A: DB2 ONLINE BACKUP ENABLEMENT

The following sections will provide an overview of DB2 backup enablement. Enablement consists of two steps.

1. Determining if database migration is required.
2. Performing the database migration for DB2 online backup enablement.

Determining if Database Migration is Required

The simplest way to determine if database migration is required is to look at some sample table definitions and inspect the logging for LOB columns. For example, the following command displays the table definition for the LONGQUESTIONRESULTS table and shows the LOB content is not logged. In this case, database migration is required based on the “NOT LOGGED” qualifier for the table’s LOB content.

```
[db2inst1@blade13 ~]$ db2look -d bfent -t longquestionresults -e
-- No userid was specified, db2look tries to use Environment variable USER
-- USER is: DB2INST1
-- The db2look utility will consider only the specified tables
-- Creating DDL for table(s)
-- This CLP file was created using DB2LOOK Version "10.5"
-- Timestamp: Mon 01 Aug 2016 06:40:15 AM CEST
-- Database Name: BFENT
-- Database Manager Version: DB2/LINUXX8664 Version 10.5.4
-- Database Codepage: 1208
-- Database Collating Sequence is: IDENTITY
-- Alternate collating sequence(alt_collate): null
-- varchar2 compatibility(varchar2_compat): OFF


CONNECT TO BFENT;

------------------------------------------------
-- DDL Statements for Table "DBO     "."LONGQUESTIONRESULTS"
------------------------------------------------


CREATE TABLE "DBO     "."LONGQUESTIONRESULTS"  (
                  "SITEID" INTEGER NOT NULL ,
                  "ANALYSISID" INTEGER NOT NULL ,
                  "PROPERTYID" INTEGER NOT NULL ,
                  "COMPUTERID" INTEGER NOT NULL ,
                  "ISFAILURE" SMALLINT NOT NULL ,
                  "ISPLURAL" SMALLINT NOT NULL ,
                  "RESULTSCOUNT" INTEGER NOT NULL ,
                  "RESULTSTEXT" CLOB(2147483647 OCTETS) NOT LOGGED NOT COMPACT NOT NULL )
                 IN "USERSPACE1" LONG IN "TSULOB"
                 ORGANIZE BY ROW;
```

Figure 22: BigFix Database LOB Logging Check

Performing the Database Migration

To perform the database migration, the following steps are recommended. Note the steps are required for the BFENT and BESREPOR databases. The sample provided below uses BFENT.

1. Stop the BigFix services.
2. In order to verify the BigFix services are indeed stopped, and not persisting connections, it can be useful to restart the database (i.e. db2stop, followed by a db2start). In the event the stop is not successful, verify the BigFix services are down and possibly force the stop.
3. Take a full offline backup of the BigFix BFENT database.

```
db2 backup db BFENT to /home/db2inst1/LOBMigration compress
```

Figure 23: Sample Database Backup with Compression Command

4. Connect to the BFENT database.

```
db2 connect to BFENT
```

Figure 24: Sample Database Connect

5. Perform the database migration step.
   Note the BigFixLOBLogging.sql script is available from the author of this paper.

```
db2 –tvf BigFixLOBLogging.sql
```

Figure 25: Sample Migration

6. Upon successful completion of the migration, it is recommended to take another offline backup of the database. In the event errors are encountered, whether as part of the migration process or once the BigFix services are started, the backup captured in the first step may be restored to reset the server state. A sample restore command follows.

```
restore db BFENT from /home/db2inst1/LOBMigration taken at 201607281234 without
prompting
```

Figure 26: Sample Database Offline Backup Restore

7. Restart the BigFix services.

Outline of Database Migration Steps

The database migration script (BigFixLOBLogging.sql) performs the following steps.

1. Creates a set of three tables (essentially temporary tables) to keep track of the LOB migration content:
   DBO.LOB_TABLES
   DBO.LOB_COLUMNS
   DBO.LOB_COLUMN_DUMP
2. Creates a set of three stored procedures to perform the migration steps:
   DBO.getLobColumns
   DBO.generateRowIDValues
   DBO.changeLOBLogging
3. The changeLOBLogging procedure is the main processing loop for the migration. It derives the set of tables that require migration from the system catalogs. For each relevant table/column the procedure will.
   - Dump the column data into one of the temporary tables.
   - Drop the column.
   - Recreate the column, with suitable logging.

- Restore the column data.
- Perform any necessary table reorganization as determined by the migration.

REFERENCES

BigFix Capacity Planning Guide

MX Performance Toolkit for BigFix

BigFix 9.5 Knowledge Center

MS SQL Transact-SQL Reference

MS SQL: Monitor and Tune for Performance

DB2 10.5 Knowledge Center

DB2: Best practices tuning and monitoring database system performance

Notices

This information was developed for products and services offered in the U.S.A.

HCL may not offer the products, services, or features discussed in this document in other countries. Consult your local HCL representative for information on the products and services currently available in your area. Any reference to an HCL product, program, or service is not intended to state or imply that only that HCL product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any HCL intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-HCL product, program, or service.

HCL may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to HCL TECHNOLOGIES LIMITED email: products-info@hcl.com

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: HCL TECHNOLOGIES LIMITED PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. HCL may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-HCL Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this HCL product and use of those Web sites is at your own risk.

HCL may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact HCL TECHNOLOGIES LIMITED email: products-info@hcl.com

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by HCL under terms of the HCL License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-HCL products was obtained from the suppliers of those products, their published announcements or other publicly available sources. HCL has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-HCL products. Questions on the capabilities of non-HCL products should be addressed to the suppliers of those products.

All statements regarding HCL's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All HCL prices shown are HCL's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

HCL, and other HCL graphics, logos, and service names including "hcltech.com" are trademarks of HCL. Except as specifically permitted herein, these Trademarks may not be used without the prior written permission from HCL. All other trademarks not owned by HCL that appear on this website are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by HCL.

IBM and other IBM graphics, logos, products and services are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Oracle database, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

VMware's and all VMWare trademarks and logos are trademarks or registered trademarks in the United States and certain other countries.

Dell, EMC, DellEMC and other trademarks are trademarks of Dell Inc. or its subsidiaries in the United States and certain other countries.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo and JBoss are registered trademarks of Red Hat, Inc. in the U.S. and other countries. Linux is a registered trademark of Linus Torvalds. All other trademarks are the property of their respective owners.

Mozilla and all Mozilla trademarks and logos are trademarks or registered trademarks in the United States and certain other countries.

Google LLC All rights reserved. Google and the Google Logo are registered trademarks of Google LLC.

NETAPP, the NETAPP logo, and the marks listed at www.netapp.com/TM are trademarks of NetApp, Inc.