

HP Business Availability Center

for the Windows and Solaris operating systems

Software Version: 8.01

TransactionVision Planning Guide

Document Release Date: March 2009

Software Release Date: March 2009



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Third-Party Web Sites

HP provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. HP makes no representations or warranties whatsoever as to site content or availability.

Copyright Notices

© Copyright 2000 - 2009 Hewlett-Packard Development Company, L.P.

Trademark Notices

TransactionVision® is a registered trademark of the Hewlett-Packard Company.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The OpenGroup.

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP Software Support web site at:

<http://www.hp.com/go/hpsoftwaresupport>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software Support Online provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the HP Software Support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Table of Contents

Chapter 1: Introduction to TransactionVision Planning	11
Deployment Roadmap	12
Deployment Planning	13
Quick Tips for a Smooth Deployment	17
Assembling the TransactionVision Team.....	19
Chapter 2: Analyzer Host Sizing	23
Define the Scope.....	23
Sizing	26
Selecting TransactionVision Pilot Hardware & Software	27
Chapter 3: DBMS Configuration, Sizing and Tuning	29
Disk Storage Requirements.....	29
DBMS Performance.....	30
Chapter 4: WebSphere Configuration	37
WebSphere MQ Configuration	37
WebSphere MQ Capacity Planning.....	38
Chapter 5: Performance Testing and Tuning	43
Setting Objectives to Optimize Performance	43
TransactionVision's Process Steps	44
TransactionVision Performance Test	46
TransactionVision Performance Tuning	49
Chapter 6: Administration	51
Maintenance and Administration.....	51
Administration Tasks.....	52
Chapter 7: Customizing TransactionVision	53
TransactionVision Customizations	53
Index.....	57

Table of Contents

Welcome to This Guide

Welcome to the TransactionVision Planning Guide. This guide contains important information for sizing and planning new installations of TransactionVision.

Note to HP Software-as-a-Service customers: The information in this guide is not relevant to HP Software-as-a-Service customers.

This chapter includes:

- ▶ Who Should Read This Guide on page 7
- ▶ TransactionVision Documentation on page 8
- ▶ Additional Online Resources on page 9
- ▶ Documentation Updates on page 10

Who Should Read This Guide

This guide is for the following users of TransactionVision:

- ▶ Application developers or configurators
- ▶ System or instance administrators
- ▶ Database administrators

Readers of this guide should be moderately knowledgeable about enterprise application development and highly skilled in enterprise system and database administration.

TransactionVision Documentation

TransactionVision documentation provides information on using the TransactionVision application of the Business Availability Center and deploying and administering the TransactionVision-specific components in the Business Availability Center deployment environment.

The TransactionVision documentation includes:

- ▶ The *TransactionVision Deployment Guide* describes the installation and configuration of the TransactionVision-specific components in the Business Availability Center deployment environment. This guide is available as a PDF in the Business Availability Center Online Documentation Library.
- ▶ The *Using TransactionVision Guide* describes how to set up and configure TransactionVision to track transactions and how to view and customize reports and topologies of business transactions. This guide is available as the TransactionVision Portal or as a PDF in the Business Availability Center Online Documentation Library.
- ▶ The *TransactionVision Planning Guide* contains important information for sizing and planning new installations of TransactionVision.
- ▶ The *TransactionVision Advanced Customization Guide* contains information for how the TransactionVision platform can be extended and customized to achieve further control over its various functions. It presents an architecture overview of the TransactionVision system and documents the different methods available to use and extend the Analyzer, the query service and the TransactionVision user interface.

Additional TransactionVision documentation can be found in the following areas of the Business Availability Center:

Readme. Provides a list of version limitations and last-minute updates. From the HP Business Availability Center DVD root directory, double-click `readme80.html`. You can also access the most updated readme file from the HP Software Support Web site.

What's New. Provides a list of new features and version highlights. In HP Business Availability Center, select **Help > What's New**.

Online Documentation Library. The Documentation Library is an online help system that describes how to work with HP Business Availability Center and the TransactionVision application. You access the Documentation Library using a Web browser. For a list of viewing considerations, see "Viewing the HP Business Availability Center Site" in chapter 6 of the *HP Business Availability Center Deployment Guide* PDF.

To access the Documentation Library, in HP Business Availability Center, select **Help > Documentation Library**. Context-sensitive help is available from specific HP Business Availability Center pages by selecting **Help > Help on this page** and from specific windows by clicking the Help button. For details on using the Documentation Library, see "Working with the HP Business Availability Center Documentation Library" in Platform Administration.

Additional Online Resources

Troubleshooting & Knowledge Base accesses the Troubleshooting page on the HP Software Support Web site where you can search the Self-solve knowledge base. Choose **Help > Troubleshooting & Knowledge Base**. The URL for this Web site is <http://h20230.www2.hp.com/troubleshooting.jsp>.

HP Software Support accesses the HP Software Support Web site. This site enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > HP Software Support**. The URL for this Web site is www.hp.com/go/hpsupport.

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:
http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport user ID, go to:
<http://h20229.www2.hp.com/passport-registration.html>

HP Software Web site accesses the HP Software Web site. This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help > HP Software Web site**. The URL for this Web site is www.hp.com/go/software.

Documentation Updates

HP Software is continually updating its product documentation with new information.

To check for recent updates, or to verify that you are using the most recent edition of a document, go to the HP Software Product Manuals Web site (<http://h20230.www2.hp.com/selfsolve/manuals>).

1

Introduction to TransactionVision Planning

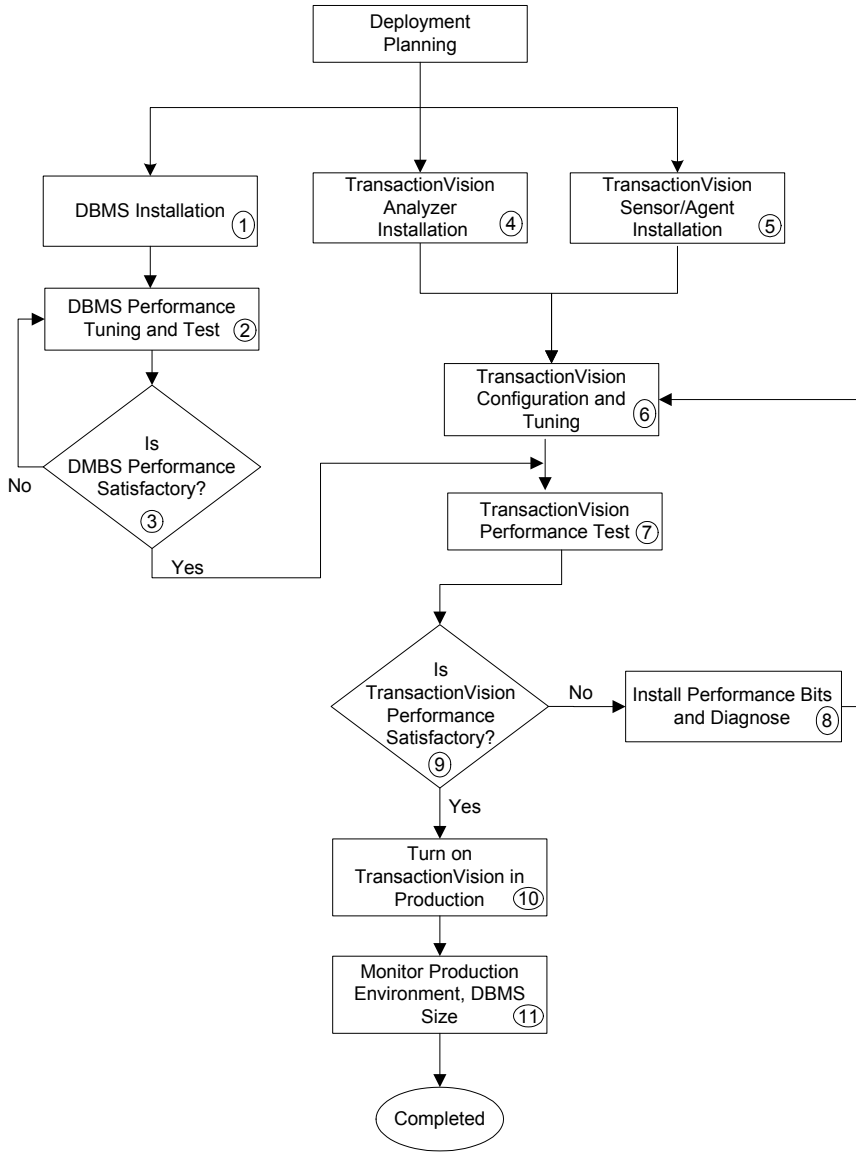
HP recommends that you plan your TransactionVision deployment carefully before installing TransactionVision. It is important to install TransactionVision components on the appropriate hosts in your environment. This chapter provides a roadmap and guidelines for planning your deployment.

This chapter includes:

- Deployment Roadmap on page 12
- Deployment Planning on page 13
- Self-Discovery of Existing Transaction Paths on page 16
- Quick Tips for a Smooth Deployment on page 17
- Assembling the TransactionVision Team on page 19

Deployment Roadmap

The following diagram shows a high level task flow depicting a typical TransactionVision deployment scenario. The numbered elements are referenced in the table on the following page, which provides additional details about the steps and a reference to more information.



The following tables describes the numbered elements in the task flow.

Ref. No.	Comment
1 - 3	See “DBMS Configuration, Sizing and Tuning” on page 29
4	See “Analyzer Host Sizing” on page 23.
5	See the <i>HP TransactionVision Deployment Guide</i> PDF.
6	See “WebSphere Configuration” on page 37.
7 - 9	See “Performance Testing and Tuning” on page 43.
10	See the <i>HP TransactionVision Deployment Guide</i> PDF.
11	See “Administration” on page 51.

Deployment Planning

Proper planning is important for successfully deploying TransactionVision, and the amount of planning necessary depends on the nature of deployment. Large production deployments require more up-front effort in capacity and resource planning, while smaller development and QA deployments with a single machine environment may require minimal planning.

Consider the following topics for deployment planning:

- “Overall monitoring strategy” on page 13
- “Deployment strategy” on page 14
- “Identifying Servers to Monitor” on page 14
- “Dependent Software Installation” on page 16
- “Self-Discovery of Existing Transaction Paths” on page 16

Overall monitoring strategy

- What applications and objects need to be monitored?
- Where are the Agents or Sensors going to be deployed?

- What communication links need to be created between the Agent or Sensor and the Analyzer?
- How much data needs to be collected at each monitoring point?
- What is the transaction rate?
- What is the peak transaction rate?
- What is the total amount of data that needs to be retained in the database before archiving/deletion takes place?
- Can TransactionVision data be partitioned? If so, what is the criteria?
- What is the level of tolerance regarding to loss of TransactionVision data?

Deployment strategy

- Where will the TransactionVision Analyzer(s) be hosted?
- Where will the DBMS be hosted?
- Personnel resource availability to install third-party software TransactionVision depends upon, such as IBM DB2, Oracle, SQL Server on any new hardware.
- Resource availability to install and setup TransactionVision.
- What are the reports to be run and how often they will be run?
- What access authorizations are to be provided to users for administering and operating TransactionVision and accessing TransactionVision data?

Identifying Servers to Monitor

TransactionVision can monitor a server running one or more of the following applications:

- WebSphere MQ applications
- JMS applications, Servlets, Enterprise Java Beans, JDBC
- CICS applications
- BEA Tuxedo applications
- NonStop TMF applications

- ASP.NET applications
- WebSphere MQ IMS

The following table provides a worksheet to help you plan and list out the servers to be monitored by TransactionVision. This information is optional, but may be useful for your capacity planning and recording purposes. Initial prototype deployments may not need to record this information. You will not be prompted to enter this information into TransactionVision during installation.

Server Host Name	Agent/Sensor Type (WMQ/WMB/ JMS/ Servlet/ EJB/JDBC/ CICS/IMS/ NonStop/.NET/ Tuxedo)	Queue Manager(s)	Client/Server Connection	z/OS CICS/ IMS Region
<p>Note: To make it easier to collect this information, you can copy the table headings into an Excel spreadsheet.</p>				

Use this table to record the number of servers where the TransactionVision Sensor has been installed, which Agent or Sensor has been installed (WebSphere MQ, WebSphere Message Broker, JMS, Servlet, EJB, CICS, IMS, NonStop Guardian, .NET, BEA Tuxedo), the queue managers the Sensor connects to, whether a client or server connection is used, and if z/OS is the platform, which CICS or IMS region the Sensor is installed under.

Additional information such as application name and WMB broker name also may be collected.

Dependent Software Installation

TransactionVision requires IBM DB2, Oracle, or SQL Server database software. Additionally, if you do not want to use the built-in SonicMQ as the messaging middleware provider, then TIBCO EMS or WebSphere MQ must be provided. See chapter 2 in the *HP TransactionVision Deployment Guide* PDF for the required versions of these software packages.

TransactionVision uses the DBMS extensively for its data collection and analysis. Hence, the performance of the DBMS is vital to the overall performance of TransactionVision.

For details on DBMS configuration and tuning, see the *HP TransactionVision Deployment Guide* PDF.

Self-Discovery of Existing Transaction Paths

In many cases, a prototype TransactionVision deployment may be used to self-discover existing transaction paths, which may help to determine the scope and capacity of the installation. The basic procedure for a prototype deployment is as follows:

- 1** Install the TransactionVision Agents or Sensors on the servers running the major applications of the system.
- 2** Set up communication links from the Agents or Sensors to the Analyzer.
- 3** Run the applications with the Agent or Sensor enabled, with data collection filters set to collect all data, for an appropriate length of time to collect all necessary data.
- 4** Use the data collected by TransactionVision to estimate message volumes, data sizes, number of queues and queue managers, application names, etc. This data can then be used to size Analyzer server hardware.

See the the *HP TransactionVision Deployment Guide* PDF and the *Using TransactionVision Guide* for information about these tasks.

Quick Tips for a Smooth Deployment

When planning your deployment, pay attention to the following tips for TransactionVision, TransactionVision on z/OS, and the DBMS.

TransactionVision

- 1** Don't undersize! Early in your planning process, determine the sizing of processor(s), processor speeds, system memory, log file(s) sizes (especially database transaction log file size) and storage space. Improper sizing could cause moderate to severe issues. Use the **TVBlast** test (see "TransactionVision Performance Test" on page 46) before installing TransactionVision in each and every environment. Contact HP to perform a proper and complete sizing exercise in all these areas with you.
- 2** Disable the **Retry Event Package Delivery on Failure** option for the communication link. Set the Sensor **put event failure retry** option to an appropriate value according to the event collection strategy. For the safest setting, uncheck the **Retry Event Package Delivery on Failure** option. The option is set on by default to ensure that you don't lose any event messages. The side effect of this is that if you don't plan your environment properly, it could clog the event queue, fill up queue manager storage space, slow down your applications, etc. Setting the option to off is the safest option. However, this does leave open the possibility of not always getting every event message.
- 3** Configure filters correctly. Data collection filters assigned to communication links should have only the required MQ objects, application, hosts and APIs specified for collecting. MQGET calls with NO_MSG_AVAIL reason code should be filtered out. When monitoring WMQI brokers, make sure to filter out API calls to WMQI system queues MQGET with NO_MSG_AVAIL reason code.
- 4** Set the **Configuration Message Expiry** in the Analyzer to a small value. This option should be set to a value small enough so that the amount of events generated by an "orphaned" configuration message can fit into the event queues without causing major production issues.

- 5 Set event queue depth/storage correctly. Event queue depth should be set to a value that matches the queue manager storage spaced based on the average event message size. It should be adequate to handle peak volume. It should not exceed the storage capability of the queue manager host system. You may want to consider having a new queue manager created for the TransactionVision Analyzer. To be certain of storage, plan and execute for maximum event message space. This is important especially when the event queue and TransactionVision dead letter queue are hosted by the production queue manager as running out of storage space may stop the queue manager completely.
- 6 Protect the production environment. A fail-safe method (such as running a **crontab** job) should be devised to make sure that if any of the TransactionVision components malfunction (such as raw event backlogging) the production environment will be protected.
- 7 Configure the Sensor or Agent's **Event Packaging** factor correctly. Event packaging factors should be set to a value suitable to the environment. This factor is usually set to 10 events per message.
- 8 Modify the **inetd** configuration file correctly. Incorrect configuration of **inetd** is the most common cause for not being able to establish new MQ client connections. Making sure the configuration is correct (including spelling!) avoids this issue.

TransactionVision in a z/OS Environment

Runtime support for language environment must be included in affected CICS regions. The language environment must be enabled in the CICS region for C or the MQ listener will be disabled or possibly MQ connections will be disabled.

Databases

- 1 Run database performance tests! Most databases are tuned for reading without much attention paid to the database's write rate (insert rate). TransactionVision is very insert intensive. Run HP's database tests before installing TransactionVision on each and every environment (see “DBMS Configuration, Sizing and Tuning” on page 29).
- 2 Set thread count accurately. Analyzer thread count should be set to match the test results from the DBMS insert rate.
- 3 Perform regular database performance maintenance. DBMS performance maintenance such as **RUNSTATS** (for DB2), or something similar for other databases, needs to be run on a regular basis (recommended daily) to ensure optimal performance.

Assembling the TransactionVision Team

Assembling the TransactionVision team has two aspects:

- “Project Skills and Organization” on page 19
- “Participant Roles” on page 21

Project Skills and Organization

In order to achieve the project objectives, HP proposes forming a project team to evaluate TransactionVision. It is recommended that individuals with skills in the following areas be available during the project to assist with installation and configuration:

- Operating system on which the TransactionVision Sensors will be installed
- WebSphere MQ network configuration used by the TransactionVision environment
- IBM DB2 UDB Enterprise Edition or other supported database

The following additional skills may also be needed during installation and configuration:

- Application server technology
- Java Beans

During the evaluation phase, individuals representing the following skill sets should participate:

- Technical Architecture
- Middleware Development & Support
- Application Development & Support
- Business Process Analysis
- ROI Analysis
- Data Security / Information Integrity
- Java Programming
- Project Management

In large-scale projects, HP recommends assigning a project manager responsible for the overall completion of the project. Other on-call skills may be required if challenges arise.

The HP Team stands ready to supplement and assist your staff concerning any of these required skill sets. HP will also provide performance-tuning assistance throughout the project.

Participant Roles

HP utilizes a project team distribution list to keep everyone informed of project issues. Identify the names, roles and contact information for individuals who will participate in the program. Be sure to also include the names of all individuals who must approve funding.

Participant's Role	Name	Title/ Position	Phone	E-mail
Project Leader				
Middleware Architect				
Enterprise Architecture				
Production Support				
Application Development				
Data Security				
Database Administrator				
WebSphere Developer				
Business Representatives				
Executive Sponsor				
Sales Engineer				
Support				
Note: To make it easier to collect this information, you can copy the table headings into an Excel spreadsheet.				

2

Analyzer Host Sizing

This chapter includes:

- Define the Scope on page 23
- Sizing on page 26
- Selecting TransactionVision Pilot Hardware & Software on page 27

Define the Scope

HP has found "scope creep" to be the number one issue facing most technology initiatives. It is therefore critical that scope be well defined at the beginning of the project. HP has identified several key questions that facilitate an enhanced understanding and definition of scope.

Scoping Questionnaire

Begin by understanding the platforms and applications that use WebSphere MQ. Complete the following table for all WebSphere MQ applications on distributed platforms:

Application Name	Server Name	Server Mfg/ Model	Operating System	# CPUs	Optional List of Queue Managers	Does The App Utilize WMQI? (Y/N)

Note: To make it easier to collect this information, you can copy the table headings into an Excel spreadsheet.

Complete the following table for all WebSphere MQ applications on mainframe platforms:

Application Name	CICS	IMS	OTMA Bridge?	Batch	Is WebSphere MQ Usage Based Pricing (MULC) utilized?	Optional List of Queue Mgrs	Does The App Utilize WMQI? (Y/N)

Note: To make it easier to collect this information, you can copy the table headings into an Excel spreadsheet.

The information provided in the previous tables represent the total universe of platforms with processes that can be tracked by TransactionVision.

The next step is to understand which set of applications form a business process that can readily be utilized in a pilot. Identify your business processes and their associated applications in the following table:

Business Process Name	Application Name	Server or Mainframe Name

Note: To make it easier to collect this information, you can copy the table headings into an Excel spreadsheet.

With the potential pilot business processes in mind, consider the following issues.

- What is your organization's current and planned middleware architecture and how does it compare to the available pilot applications/processes?
- Will MQSI be used in production or during the pilot?
- What is the availability of key staff?
- What role should the business community play?

- What is the end-to-end transaction time today? What message rate should be set as a hurdle? Does the hurdle rate need to be reached during the pilot? Can the pilot environment support the required message rate?
- Can a few critical platforms be used for the pilot, or must every potential platform be evaluated?
- Can one business process be used for the pilot?
- Does the pilot environment have sufficient hardware/software to conduct the test?
 - Does the pilot environment have the prerequisite software?
 - How many messages must be captured and stored during the test? Does the pilot environment have enough disk space to store the messages?

Final considerations should include the hardware requirements for the pilot. HP can provide a hardware sizing recommendation for the pilot based on the information provided in the Sizing section. After carefully considering these issues, you will be prepared to make an informed selection of the pilot business processes.

After a pilot environment is selected, an overall schematic of the application architecture should be drawn and a description of the process should be documented.

Sizing

HP can determine sizing for the pilot environment based on the information provided in the following table:

Data Partition (Business Process Name)	Estimated Average # of End-to-End Business Transactions per Day	Estimated Peak End-to-End Business Transactions per Day	Estimated Total # of API Calls per End-to-End Business Transaction to Be Monitored	Estimated Average Message Size including User Buffer

Note: To make it easier to collect this information, you can copy the table headings into an Excel spreadsheet.

HP can also provide sizing for the complete roll-out if you provide the information requested in this table regarding the production environment.

In a pilot environment a single instance of the TransactionVision Analyzer is usually deployed. In a production environment, however, deploying several Analyzers might be advantageous. Deploying several Analyzers requires partitioning data. In a partitioned environment, the information requested in the above table should be provided for each data partition.

There are two alternatives for partitioning data.

- 1 TransactionVision's data collection filters can be used to partition data. In order to take advantage of TransactionVision's filters, events captured by TransactionVision need to contain enough information to enable the data filters to partition data. For example, the MQ header often contains information that can be used to identify specific business applications. Organizations are encouraged to explore this alternative more fully with HP Software.
- 2 Alternatively, an organization's system architecture can be used to partition data. Many system architectures require a different message route for each business application. In these environments, data can be partitioned based on the company-wide system architecture.

A key aspect of sizing for the TransactionVision Analyzer is the number of messages per second and the average message size, including the user buffer. The following table is used as one factor to calculate the sizing for your environment:

Benchmarks with 1,000,000 messages containing 1,024 bytes of user data generated this TransactionVision Analyzer throughput model for the HP DL580 ProLiant Server.	
1,600	messages per second
96,000	messages per minute
5,760,000	messages per hour
138,240,000	messages per day
Hardware environment: 4-way Quad Core HP DL580 G5 ProLiant Server (1.6 GHz CPUs)	
Software environment: Linux RH Enterprise 4, DB2 EE 9.2	

Note: Performance is affected by factors such as the type of messages collected, the payload size, and any custom correlation. Each environment is different and may exhibit different results than shown above.

Selecting TransactionVision Pilot Hardware & Software

Determining the appropriate platforms for TransactionVision requires an understanding of TransactionVision's major components: the Sensors or Agent, the UI/Job Server and the Analyzer.

The TransactionVision UI/Job Server is a web application running on an instance of the Apache Tomcat Servlet/JSP container. The Server communicates with the Analyzer to provide data collection configuration information such as communication links and data collection filters. It also connects to project database schemas to display project analysis and report results.

The TransactionVision Analyzer is a service on Windows (or a daemon on UNIX) that communicates with TransactionVision Sensors via messaging middleware. It generates and delivers configuration messages to Sensors by placing them on a designated configuration queue. Configuration messages specify Sensor configuration information such as the name of the event queue where the Sensor should place event messages and data collection filter definitions for the project.

By default, TransactionVision uses SonicMQ as the messaging middleware provider. TIBCO EMS and WebSphere MQ are also supported.

The Analyzer also retrieves events placed on an event queue by Sensors and processes them for analysis and display by the web user interface. It performs the unmarshalling, correlation, analysis, and data management functions.

TransactionVision Sensors collect transactional events from the various applications involved in your distributed transactions. Sensors are lightweight libraries or exit programs that are installed on each computer in your environment. The Java Agent combines the capabilities of the TransactionVision Java Sensors (JMS, Servlet, JDBC and EJB) and the Diagnostics Java Probe into a single component. Similarly, The .NET Agent combines the capabilities of the Diagnostics .NET Probe and the TransactionVision .NET Sensor into a single component.

The hardware requirements for the Analyzer and UI/Job Server components increase as the number of messages and users increases. In a limited pilot, all components can usually run on a single Windows or Linux workstation that is fast. A specific sizing recommendation can be provided based upon the volume information provided in the tables completed during the scope step.

See the *HP TransactionVision Deployment Guide* PDF for information about the TransactionVision components.

3

DBMS Configuration, Sizing and Tuning

Because TransactionVision uses the DBMS extensively for its data collecting and analyzing process, the performance of the DBMS is vital to the overall performance of TransactionVision. Inserting records and updating records represent the majority of the database operations associated with TransactionVision; therefore the speed of the physical disks/I/O interface has a significant impact on the performance.

This chapter includes:

- Disk Storage Requirements on page 29
- DBMS Performance on page 30

Disk Storage Requirements

One factor that needs to be finalized is the amount of disk storage space required for TransactionVision events. This is estimated by the average size of the monitored application messages, the rate of events collected by TransactionVision, and the duration of which TransactionVision event data needs to be kept in the database.

The formula for calculating disk storage usage (for DB2) is:

$$\text{(Average message size + 11K Byte) * Event Rate * Number of Collection Points * Event Retention Time = Storage Size}$$

For example, if the average message size is 2KB, the transaction rate is 5 transactions/second (for 8 hours/day and all weekdays, this translates into 720 thousand transactions/week). If there are two collecting points for each transaction and if TransactionVision data is required to be stored for duration of four weeks before the data is either archived or deleted, then the total required storage is about 90GB

$$\text{((2 + 11)K Bytes * 720,000 * 2 * 4 = 75G Bytes).}$$

Using DB2 compact LOB feature and Analyzer compression function may reduce this number by 30% - 70%.

Also note that the application message (user data) can be filtered out using data collection filters at each collection point, and be collected at only one of the collection points in order to have the message available for the business transaction. This also helps reduce disk space requirements.

DBMS Performance

The key to DBMS performance is to overcome the operation bottleneck—I/O throughput limit. Usually this limit is imposed by the physical disk and the I/O interface.

One of the first things in deployment is to make sure the actual DBMS system has a good I/O and disk subsystem attached and that the subsystem has been tuned for writing. This includes checking that the disk is RAID configured for performance, write-cache is enabled for the disks, and the I/O interface is fast (preferably fiber-optic interface).

To achieve real high throughput of I/O, some forms of parallel processing should be used:

- ▶ Use separate DBMS instances for separate projects - if the data can be partitioned then separate DBMS instances on different hosts can be employed to achieve parallelism. This setup requires setting up multiple instances of TransactionVision.
- ▶ Use RAID disks for table space containers. RAID disks provide parallel I/O at hardware level.
- ▶ Separate table space containers and log file directories. Log files (DB2 term, Rollback Segment for Oracle) holds uncommitted database operations and usually is highly utilized during database insert/update. For this reason it should have its own container on physically separated disks, and preferably on RAID disks.
- ▶ Stripe table spaces. If parallel I/O is not available at the hardware level, DBMS can be set up to span a tablespace across multiple disks, which introduces parallelism at the DBMS space management level.
- ▶ Large-scale deployment may require an even higher degree of parallelism at the DBMS level, such as using DB2 EEE (Parallel Edition) database.

There are many other database parameters that may impact the performance of TransactionVision and need to be examined one-by-one to make sure they are optimized to the specific DBMS system.

There is also some benefit when the tablespaces used by TransactionVision is managed by the database directly (DMS or DMS RAW tablespaces).

Testing DBMS and Diagnosing Performance bottleneck

HP provides an independent tool (DB2Test, OracleTest, SQLServerTest) that can be used to test the performance of DBMS relevant to TransactionVision (especially the record insert rate). The tool is written in Java and should be run on the host on which the TransactionVision Analyzer will be installed.

The tool simulates the database insert operations generated by TransactionVision. The test should be run multiple times to get a complete picture of the DBMS performance. note that the result of the test does not directly correlate with TransactionVision processing rate, rather it is an indicator of how well does the DBMS performance for the given configuration.

Make sure each test lasts at least a few minutes to minimize the overhead of any initialization process. The test should be run against the same database and table sets with which the TransactionVision Analyzer will be run.

To run the test:

- 1** Run the insert test with one thread and with record size of 1KB; this will gauge the raw event insert performance.
- 2** Run the insert test with multiple threads and with a record size of 1KB. This will test whether the insert can benefit from multiple threads. Usually the thread count is set to two times the number of CPUs.
- 3** Run the insert test with one thread and with record size of (7KB); this will gauge the analyzed event insert performance.
- 4** Run the insert test with multiple threads and with record size of (7KB). This will test if the insert can benefit from multiple threads. Usually the thread count is set to two to four times the number of CPUs.
- 5** If the Analyzer host and the DBMS host are different, the above tests should be run on the Analyzer host. However at least one test should be run on the DBMS host to see if there is any communication/DB client configuration related issues.

The rate of insert should be on par with the result achieved from similar systems tested by HP. During the test the following parameters of the DBMS system should be monitored:

- 1** Disk I/O usage for all involved physical disks (tablespaces and log files), especially I/O busy percentage.
- 2** CPU usage, including wait time percentage.

If a disk hits 80-90% utilization or the I/O wait time is extraordinary long, that's an indication of a disk I/O bottleneck. If no obvious bottleneck is seen, then a DBMS configuration or O/S configuration issue may exist. Check DB, DBMS and kernel parameters with HP for any configuration issues.

Another useful tool for analyzing DBMS performance is the DB2 performance snapshot.

Database Test

The database tests use the following utilities provided by TransactionVision:

- DB2Test
- OracleTest
- SQLServerTest

Examples of how to run the tests are given below. For information about the syntax and full description of the utilities, see Appendix A in the *HP TransactionVision Deployment Guide* PDF.

Prerequisites

- Java Runtime Environment 1.5 or above
- DBMS server or client installed
- JDBC 2.0 driver is set in the CLASSPATH environment variable. For DB2 the driver is db2java.zip. For Oracle the driver is classes12.jar. For SQL Server, the following files must be added to the CLASSPATH:
 - msbase.jar
 - mssqlserver.jar
 - msutil.jar

A test structure is set up in the database. Use the SQL statements in the following database-specific sections to create the database structure.

DB2 Example

```
CREATE TABLE SCHEMA_NAME.RAW_EVENT (  
    event_id INTEGER NOT NULL,  
    event_status INTEGER NOT NULL,  
    event_time TIMESTAMP NOT NULL,  
    commlink_secs INTEGER NOT NULL,  
    commlink_msecs INTEGER NOT NULL,  
    client_secs INTEGER NOT NULL,  
    client_msecs INTEGER NOT NULL,  
    event_data BLOB(10M) NOT NULL,  
    overflow_id INTEGER,  
    CONSTRAINT PK_RAW_EVENT PRIMARY KEY (event_id) )  
    <IN TABLESPACE>;
```

Replace SCHEMA_NAME with a suitable schema name in the test environment. If not using the default table space USERSPACE, <IN TABLESPACE> should be specified with the actual tablespace name.

Example of running the insert test with 7K and multiple threads (4 CPUs):

```
java com.bristol.tvision.admin.DB2Test localhost tvision 50000 db2inst1 ibmdb2  
TESTSCHEMA 1000000 7000 8 -RAW_EVENT -LOB -commit 50 -jdbcBatch
```

Oracle Example

```

CREATE USER "SCHEMA_NAME"
  PROFILE "DEFAULT" IDENTIFIED BY "SCHEMA_NAME"
  DEFAULT TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP"
  QUOTA UNLIMITED ON USERS ACCOUNT LOCK;
GRANT "CONNECT" TO "SCHEMA_NAME";
CREATE TABLE SCHEMA_NAME.RAW_EVENT (
  event_id INTEGER NOT NULL,
  event_status INTEGER NOT NULL,
  event_time DATE NOT NULL,
  commlink_secs INTEGER NOT NULL,
  commlink_msecs INTEGER NOT NULL,
  client_secs INTEGER NOT NULL,
  client_msecs INTEGER NOT NULL,
  event_data BLOB,
  overflow_id INTEGER,
  CONSTRAINT PK_RAW_EVENT PRIMARY KEY (event_id) );

```

Replace SCHEMA_NAME with a suitable schema name in the test environment. If not using the default table space USERS, USERS should be replaced with the actual tablespace name.

Example of running the insert test with 7K and multiple threads (4 CPUs):

```

java com.bristol.tvision.admin.OracleTest localhost tvision 1521 system oracle
TESTSCHEMA 1000000 7000 8 -RAW_EVENT -LOB -commit 50 -oracleBatch -thin

```

SQL Server Example

```
exec sp_addrole 'SCHEMA_NAME'  
go  
BEGIN TRANSACTION;  
CREATE TABLE SCHEMA_NAME.RAW_EVENT (  
    event_id INT NOT NULL,  
    event_status INT NOT NULL,  
    event_time DATETIME NOT NULL,  
    commlink_secs INT NOT NULL,  
    commlink_msecs INT NOT NULL,  
    client_secs INT NOT NULL,  
    client_msecs INT NOT NULL,  
    event_data IMAGE,  
    overflow_id INT, CONSTRAINT PK_RAW_EVENT  
    PRIMARY KEY (event_id) )  
COMMIT
```

Replace SCHEMA_NAME with a suitable schema name in the test environment.

Example of running the insert test with 7K and multiple threads (4 CPUs):

```
java com.bristol.tvision.admin.SQLServerTest localhost 1433 tvision sa sql  
TESTSCHEMA 1000000 7000 8 -commit 50 -jdbcBatch
```

4

WebSphere Configuration

This chapter includes:

- ▶ WebSphere MQ Configuration on page 37
- ▶ WebSphere MQ Capacity Planning on page 38

WebSphere MQ Configuration

The following items should be checked for the WMQ objects created for TransactionVision:

- ▶ Event queue Queue manager. It is best to have the actual event queue hosted on a queue manager other than the production queue manager, so that event queue related issues (such as running out of disk storage due to event backlog) do not affect normal production functions.
- ▶ Event queue depth should be set to a value that matches the queue manager storage space based on the average event message size. It should be adequate to handle the peak volume.
- ▶ Event queue storage should match the event queue depth based on the average event message size. It should not exceed the storage capability of the queue manager host system. This is important especially when the event queue and TransactionVision dead letter queue are hosted by the production queue manager as running out of storage space may stop the queue manager completely.
- ▶ Event queue message persistency property should match the event collecting policy.

WebSphere MQ Capacity Planning

This chapter provides a methodology for determining how TransactionVision would impact the number of messages handled by a given system. Then, these adjusted numbers can be used with the information in the MQSeries Planning Guide (CSQZAB03) to calculate storage needs, etc.

Calculating TransactionVision's Impact on Message Volume

Use the following formula to determine the impact of TransactionVision monitoring on message volume:

$$\text{impact factor} = 1 + (\text{application efficiency factor} / \text{event packaging number})$$

where:

Variable	Description
impact factor	The impact of TransactionVision monitoring on message volume. This number can be multiplied times the existing message volume to show how much message volume would be increased with TransactionVision monitoring. Note that when TransactionVision is not actively being used to monitor a system, there is no impact on message volume, so this factor becomes one.

Variable	Description
application efficiency factor	A number from 1 to 5 approximating how efficiently the applications use the WebSphere MQ APIs. It is the ratio of the number of WebSphere MQ APIs called to the number of messages handled. In the worst case scenario, each MQPUT or MQGET will be accompanied by an MQCONN, MQOPEN, MQCLOSE and MQDISC. In this case, the efficiency of the application would be 5, since approximately five APIs are called for every message processed. At the other end of the spectrum, if an application were to call MQCONN, MQOPEN and then MQPUT 1,000 messages and MQCLOSE, MQDISC, the total number of APIs is 1,004. Dividing this by the number of messages processed (1,000) gives us a ratio of 1.004. Estimating this ratio is necessary to determine how many events TransactionVision will generate, and hence how many event messages TransactionVision will send.
event packaging number	The parameter set in the data collection filter telling the Sensor to pack a certain number of events into one event message. Note that if event packaging is not enabled, then this factor has a value of 1.

This impact factor may have to be calculated independently for different applications which are being monitored, since these applications may have different efficiency factors.

Consider an example. Suppose TransactionVision is being used to monitor an application running in batch mode overnight. This application runs on z/OS and reads data from some internal datasets and pushes that data out using messages to a variety of other servers. On average the application sends out 2,500 messages spread evenly over 25 different queues. The application is written efficiently - there is only one MQCONN and MQDISC needed and then an MQOPEN/MQCLOSE pair for each queue. Each batch of 1,000 messages is also committed or rolled back using MQCMIT or MQBACK.

An efficiency rating for this application can be calculated as follows:

$2,500 \text{ (msgs)} + 2 \text{ (MQCONN/MQDISC)} + 50 \text{ (MQOPEN/MQCLOSE)} + 25 \text{ (MQCMIT)} = 2577 \text{ apis}$

$2,577 \text{ apis} / 2,500 \text{ msgs} = 1.03 \text{ application efficiency factor}$

Now, apply the application efficiency factor to the number of messages sent and calculate different impact factors based how the event packaging is set.

Event Packaging Number	Impact Factor
1	$1+(1.03/1) = 2.03$
10	$1+(1.03/10) = 1.103$
25	$1+(1.03/25) = 1.0412$

Since this is a very efficient application, the TransactionVision impact can be minimized quite well with a reasonable packaging number. Simply multiply the impact factor times the number of messages processed, 2,500 in this case, and the result indicates that the total message traffic has increased to approximately 2,603 using the a packaging number of 25.

The impact factor can be used in the standard capacity planning formulae in the MQSeries Capacity Planning Guide which depend on either total messages or messages/second figures. Simply multiply the impact factor times the current messages or messages/second figure to get a new result which takes into account the increased traffic due to TransactionVision. Note that the use of this factor only accounts for increased message volume, but does not account for the message size.

Message Size

Message size has an impact on the TransactionVision data. If messages are small, only several thousand bytes for example, then the relative size of API overhead (message descriptors, put message options, etc.) tracked by TransactionVision is large. If the messages themselves are large, however, the relative size of overhead is small, but all of the content of the messages is being duplicated. In cases where very large messages (greater than 500K, for example) are sent, it may be desirable to use data ranges in the data collection filter to have TransactionVision collect only a portion of the data. This minimizes strain on the storage requirements of the WebSphere MQ system.

Data Collection Filters

The TransactionVision impact formula presented in this appendix considers the scenario where the broadest data collection filter is applied—every WebSphere MQ API is captured. The impact factor can be drastically reduced by defining more specific data collection criteria. For example, a simple data collection filter might capture only MQGET and MQPUT1 APIs. In this case, the application efficiency factor is reduced to 1, since no extraneous APIs are captured.

Since the data collection filter allows for great flexibility, it is difficult to create a formula to calculate the effect of a given data collection filter. Instead, the impact factor may be considered a worst case scenario which may be reduced through the use of appropriate data collection filters.

5

Performance Testing and Tuning

This section describes strategies to minimize the overhead on transactional performance due to the installation and active operation of TransactionVision. When appropriately configured, TransactionVision should enable you to collect all the required analytical information about your systems and still meet your transactional service levels.

This chapter includes:

- ▶ Setting Objectives to Optimize Performance on page 43
- ▶ TransactionVision's Process Steps on page 44
- ▶ TransactionVision Performance Test on page 46
- ▶ TransactionVision Performance Tuning on page 49

Setting Objectives to Optimize Performance

Most organizations establish service level standards for a complete, end-to-end-business process. This business process can include application logic, information retrieval from one or more data sources, database inserts, deletes and updates, data transformation and middleware messaging. The contribution of any one of these steps toward the end-to-end business process performance depends on the overall business process design. For example, a process that is solely responsible for routing messages relies significantly more on middleware performance than a business process that needs to access a database and performs application logic.

Business processes typically do more than just flow through middleware—they also usually involve accessing data, performing application logic and often data transformation. Therefore, performance overhead should be evaluated based on the entire end-to-end business process. Once you have identified the end-to-end business process that you need to optimize, the performance overhead of TransactionVision can be evaluated and optimized.

TransactionVision's Process Steps

When actively collecting data, TransactionVision performs three tasks after an application makes a WebSphere MQ API call:

- 1** Inspect the content of the call against the data collection filter criteria.
- 2** If the message matches the filter, TransactionVision makes a copy of the message content and calls MQPUT to send the event asynchronously back to the TransactionVision database.
- 3** Pass the complete message to the WebSphere MQ library.

After passing the call to the MQ library, the application resumes its normal flow by executing the actual API call.

Based on this sequence of events, there are two main factors that dictate the performance impact of TransactionVision: Data Collection Filters and Event Buffering.

Data Collection Filters

The Data Collection Filters determine which messages and how much of each message is saved to the database. Data Collection Filters can be set based on several parameters including: Host Name, User Name, Program Name, Time, CICS Region, CICS Transaction, Job Name, API, API Return Code, QueueManager, MQSI Broker, MQSI Message Flow, IMS Region Type, IMS Region Identifier, IMS Identifier, IMS Transaction, and IMS PBS. By tuning these parameters, filters that limit the volume of messages collected by TransactionVision are created.

The data collection filter parameters that most significantly affect system performance are related to the MQ series User Data Buffer section. TransactionVision can be configured to collect API names only, API name plus API header information or API, Header and User buffer.

Parameters can also be set to collect only critical sections of the user buffer. For example parameters can:

- ▶ Set the Data Collection Filters to collect only APIs with failure or warning return codes.
- ▶ Set Data Collection Filters to collect only the first 100 bytes of each message user buffer.

TransactionVision can even be configured to measure and identify its own performance bottlenecks. For example, TransactionVision performance can be evaluated through a post-collection Java bean that measures end-to-end transaction performance at the end-point of a request/reply system. If performance exceeds the desired service level, additional Sensors can be activated automatically to collect all the data that will help determine the source of the problem.

Event Buffering

An additional performance tuning mechanism provided by TransactionVision is event buffering. By default TransactionVision calls MQPUT each time a message matches a Data Collection Filter. The Event Packaging parameter buffers multiple messages at the Sensor and packages them into a single MQPUT call. For example, event buffering can enable TransactionVision to send one TransactionVision message for every ten application messages.

Summary

By utilizing the data collection, configuration and operational strategies discussed above, you should be able to collect all required information about your systems and meet your transactional service level objectives. Additionally, you may find that your new end-to-end transaction analysis capabilities provided by TransactionVision more than justify any incremental computer resources and capacity utilized. HP's Software Support organization stands ready to assist customers with setup and configuration to enable you to achieve maximum value and performance.

TransactionVision Performance Test

Prior to activating TransactionVision, it is recommended to run the test application provided (**tvblast**), to verify that the underlying infrastructure that is running TransactionVision can handle the volume generated by the production applications.

This test should be run in the same environment (or a mirrored environment) where the production application will run. The test result should be used to check whether the system has been tuned up to the same level of similarly configured systems.

The test should be run to generate events with the same size expected in production, and at a similar event rate. The whole event collecting and analyzing path will be tested this way. During the test, system resources across all involved platforms should be monitored for any potential issues.

Prerequisites

- WMQ server installed
- Sensor installed on the test platform (supported platforms are Windows NT 4.0/2000, Solaris, AIX and HP-UX)
- TransactionVision Analyzer installed and configured for collecting events from the test platform
- One available queue defined at the queue manager to be tested

PUT format

Syntax:

```
tvblast -c count -b message_size -W wait_interval -t trace_level -pprint_iteration -u
unit_of_work -s status_report queue_manager queue
```

Where:

Option	Description
count	Number of total messages to write.
message_size	Size of message user buffer in bytes (default is 1000).
wait_interval	Amount of time in milliseconds to wait before proceeding to the next message.
trace_level	Controls how detailed the trace information is. Usually set to 2.
print_iteration	Controls how often (in messages) the trace message is given.
unit_of_work	Controls how many messages are included for each unit of work.
status_report	Controls how often (in messages) a statistics report is generated.
queue_manager	The name of the queue manager the program connects to.
queue	The name of the queue that is to be used for the test.

GET format

Syntax

```
tvblast -c count -b buffer -r -C -W wait_interval -t trace_level -p print_iteration -u
unit_of_work -s status_report queue_manager queue
```

Where:

Option	Description
count	Number of total messages to read.
buffer	Size of the receiving buffer.
wait_interval	Amount of time in milliseconds to wait before proceeding to the next message.
trace_level	Controls how detailed the trace information is. Usually set to 2.
print_iteration	Controls how often (in messages) the trace message is given.
unit_of_work	Controls how many messages are included for each unit of work.
status_report	Controls how often (in messages) a statistics report is generated.
queue_manager	The name of the queue manager the program connects to.
queue	The name of the queue that is to be used for the test.

Note: To see the usage information for **tvblast**, enter: *** tvblast -x** .

Examples

If the production environment has a transaction rate of 5 transactions/second with two data collecting points (MQPUT and MQGET) and average message buffer size is 2000 bytes. The following test can simulate the production scenario in a 10-minute test using two tvblast instances for writing and reading:

- Message rate: 5 message/sec for each tvblast instance. Set the message wait interval to 200ms (-W 200).
- Message size: 2000 bytes. Set the buffer size to 2000 (-b 2000).

- ▶ Test duration: 10 minutes. Set the total message count to 3000 messages (-c 3000).
- ▶ Unit of work: assuming each message is in its own unit of work (-u 1).

```
tvblast -c 3000 -b2000 -t 2 -p 50 -u 1 -W 200 -s 300 test_manager test_queue  
tvblast -c 3000 -b2000 -C -r -t 2 -p 50 -u 1 -s 300 test_manager test_queue
```

Note: The two tests should be run simultaneously. The data collection filter should be set to collecting MQPUT and MQGET in general.

TransactionVision Performance Tuning

If the performance test results show that the system does not perform as expected, further investigation is required to identify the bottleneck.

The first place to look is the performance data from the test, particularly the system resource utilization data. There may be resource bottlenecks (CPU utilization on the Analyzer hosts, disk I/O utilization on the DBMS hosts, etc.) that can be immediately recognized from this data.

If there is no obvious source for the under performance, then a diagnostic test needs to be performed to break down the event processing time. A TransactionVision build with the performance diagnosis function enabled needs to be installed and the performance test needs to be repeated. TransactionVision will generate detailed performance information that is helpful in pinpointing the cause of the performance issue.

6

Administration

This chapter includes:

- Maintenance and Administration on page 51
- Administration Tasks on page 52

Maintenance and Administration

The following tasks need to be planned for ongoing TransactionVision administration:

- **DBMS storage space management.** An effective mechanism needs to be devised to control the storage space used by TransactionVision data. This may include a regular data backup and deletion schedule. A policy needs to be in place to decide what data needs to be backed up. This can be decided based on what the data is being used for. While every scenario may be different, some examples might be to save complete event data which may be used for short term diagnostics and may be preserved for just a week. Alternatively, message data may be used for auditing and may need to be preserved for years. Statistical data may need to be preserved for a long period to observe trends in volume and response time.
- **DBMS performance maintenance.** Database statistics need to be generated on a regular basis to ensure optimal performance.
- **Adding and changing projects, schemas, users, queries, filters, Sensors and communication links.** Refer to the *Using TransactionVision Guide* for details on these tasks.

Administration Tasks

The following tasks need to be planned for TransactionVision administration:

- 1** DBMS storage space management—an effective mechanism needs to be devised to control the storage space used by TransactionVision data.
- 2** DBMS performance maintenance—database statistics need to be generated on a regular basis (recommended daily) to ensure optimal performance.
- 3** TransactionVision runtime environment itself needs to be monitored to make sure things like TransactionVision log files do not run out of space.
- 4** A fail-safe method (such as running a **crontab** job) should be devised to make sure that if any of the TransactionVision components malfunction (such as raw event backlogging) the production environment is protected.

7

Customizing TransactionVision

This chapter includes:

- ▶ TransactionVision Customizations on page 53

TransactionVision Customizations

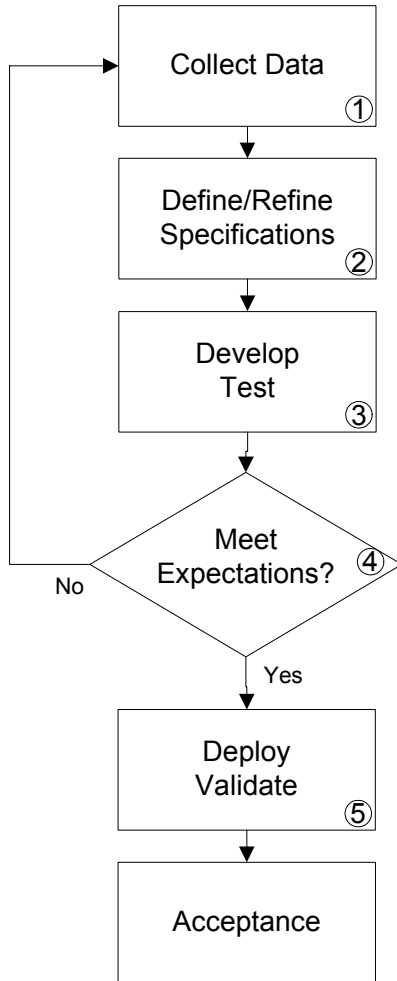
Depending on the data collected, TransactionVision can be customized in several ways. The primary customizations include:

- ▶ Writing Analyzer beans and XDM files to extract message data fields, writing custom event correlation beans or writing transaction classification rules. Often, message data contains useful information from which custom reports can be built. The first step for using this information is to convert binary message data into XML by writing an unmarshaller bean and then mapping XML fields into database table columns using TransactionVision XDM files. Event correlation beans may need to be written if parts of your system are not being monitored by TransactionVision. Transaction classification rules can be added to classify your system transactions into different categories and add attributes to those categories.
- ▶ Writing job beans to perform batch data analysis or administration tasks. Job beans may be written if you need to perform any kind of a batch job such as data backup, deletion, custom analysis and so forth.

For details on implementing the above customizations, see the *TransactionVision Advanced Customization Guide*.

- ▶ Writing custom reports. Custom reports perform analysis on data collected by TransactionVision that is specific to your system being monitored.

The flowchart below describes the process to create a custom report. The numbered elements are referenced in the table on the following page which provides additional details about the steps and references to more information.



The following tables describes the numbered elements in the task flow.

Ref. No.	Comment
1	Collect TransactionVision data from customer environment. This step usually involves installing, configuring and running TransactionVision in some customer environment. The customer environment can be development environment, QA environment or production environment, as long as the actual (or mirrored) transaction flow can be captured by TransactionVision. The captured data can be used as the base for demonstrating TransactionVision capability, developing requirement for custom reports and later testing the reports. This step may be done during the Proof-of-Concept stage.
2	Define/Refine custom report requirement specification. Once customer has better understanding of the data collected by TransactionVision and how the data can be used to solve technical and business issues, a specification shall be developed by the customer to formalize the requirement for the custom reports. The specification should cover the logic for generating the reports, detailed user interface description (possibly with screen mock-ups or samples) if applicable, the manner the reports will be invoked (scheduled, interactive, etc) and response time and any relevant topics. HP will provide an estimate for the amount of work that is involved to implement the reports.
3	Develop and test custom reports. Once the specification is finalized, work can start to implement the logic to create the reports. The work will be conducted at HP. The work may involve both front end (user interface) and back end (data analysis) programming. HP will conduct the development in house and test the reports against the data collected from customer environment.

Ref. No.	Comment
4	Review reports. Once the reports have been tested, the customer will have an opportunity to review the reports. If the reports does not meet all customer expectations, or based on the reports the customer has developed additional requirement, go back to step 2 to refine the requirement specification.
5	Deploy and validate report. Once the customer has approved the reports, they can be deployed to the customer environment for final validation. Sometimes re-configuration of the reports is necessary if the final deployment environment is different from the environment where the initial test data were collected. A user acceptance test usually is conducted after the deployment to make sure that both the content of the reports and the response time of the reports are acceptable to the customer.

Index

A

Analyzer

- benchmarks 27
- daemon 28
- defining scope 23
- hardware requirements 28
- service 28
- sizing 27

- applications that can be monitored 14
- audience, for documentation 8

C

- Configuration Message Expiry option 17
- conventions, typographical 10

D

database, see DBMS

DBMS

- maintenance 19
- performance 29

deployment strategy 14

disk storage requirements 29

documentation

- download site 10
- overview 8

E

Event Packaging factor 18

H

hardware requirements 28

HP Software Support Web site 9

HP Software Web site 10

I

inetd configuration file 18

J

Java Agent, purpose 28

K

Knowledge Base 9

M

messaging middleware provider 28

N

NET Agent, purpose 28

O

online resources 9

R

Retry Event Package Delivery on Failure
option 17

S

self-discovery of existing transaction paths
16

Sensors, purpose 28

Index

T

TransactionVision

- deployment scenario 12

- documentation set 8

- team 19

TransactionVision UI/Job Server 27

Troubleshooting and Knowledge Base 9

typographical conventions 10

U

UI/Job Server

- hardware requirements 28

user roles 21