

TransactionVision®

TransactionVision Analyzer Installation and Configuration Guide *Version 5.0.0 SPF*

Printed June 16, 2006

This manual supports TransactionVision Release 5.0.0 SPF.

No part of this manual may be reproduced in any form or by any means without written permission of:

Bristol Technology Inc.
39 Old Ridgebury Road
Danbury, CT 06810-5113 U.S.A.

Copyright © Bristol Technology Inc. 2000 — 2006
Protected by U.S. Patent 7,003,781.

RESTRICTED RIGHTS

The information contained in this document is subject to change without notice.

For U.S. Government use:

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013.

All rights reserved. Printed in the U.S.A.

The information in this publication is believed to be accurate in all respects; however, Bristol Technology Inc. cannot assume responsibility for any consequences resulting from its use. The information contained herein is subject to change. Revisions to this publication or a new edition of it may be issued to incorporate such changes.

Bristol Technology® and TransactionVision® are registered trademarks of Bristol Technology Inc. The IBM e-business logo, zSeries, z/OS, S/390, OS/390, OS/400 and WebSphere MQ are all trademarks of IBM Corporation. All other trademarks herein are the property of their respective holders.

General Notice: Some of the product names used herein have been used for identification purposes only and may be trademarks of their respective companies.

Part No. TV11060509 SPF

Contents

Chapter 1	TransactionVision Analyzer Overview.....	1
	Installation Overview	2
	Upgrading from Previous TransactionVision Releases.....	3
	Additional TransactionVision Resources	3
Chapter 2	Installing the Analyzer on Distributed Platforms.....	5
	Initial Installation	6
	Upgrade Installation	6
	Installation Files	8
	Uninstalling the Analyzer.....	9
Chapter 3	Installing the Analyzer on Windows.....	11
	Initial Installation	11
	Upgrade Installation	12
	Modifying the Installation.....	13
	Uninstalling the Analyzer.....	15
Chapter 4	Configuring Databases	17
	Identifying Database Connection Names	17
	Setting DB2 Variables.....	20
	Setting Oracle Variables.....	22
	Connecting to an Oracle Database	22
	DBMS Performance Tuning.....	23
	DBMS Disk Space Requirements	27
	Configuring Databases for Unicode Data.....	28

Chapter 5	Configuring the Analyzer	29
	Configuration Overview.....	29
	Using TVisionSetupInfo.....	30
	Using TVisionSetup	38
	Additional Analyzer Configuration.....	42
	Reducing Event Database Size.....	45
Chapter 6	Configuring Analyzer Logging.....	47
	Log Files.....	47
	Circular Logging	47
	Trace Logging	49
	Using Windows and UNIX System Logs.....	49
	Enabling SMTP Logging.....	50
	Enabling SNMP Logging.....	51
Chapter 7	Integrating TransactionVision with Tivoli Enterprise Console.....	53
	TEC Integration Setup.....	53
	Installation and Setup	54
Appendix A	Utilities Reference.....	59
	CreateSqlScript.....	59
	DB2RunStats	62
	DB2Test	64
	MigrateConfig	65
	MigrateDB.....	66
	OracleRunStats.....	67
	OracleTest	69
	ServicesManager	70
	TVisionSetup.....	73
	TVisionSetupInfo	75
Appendix B	Configuration Files.....	77
	Analyzer.properties	77
	CacheSize.properties	82
	Database.properties	82
	License.properties	85
	Performance.properties.....	85
	Setup.properties.....	85

Chapter 1

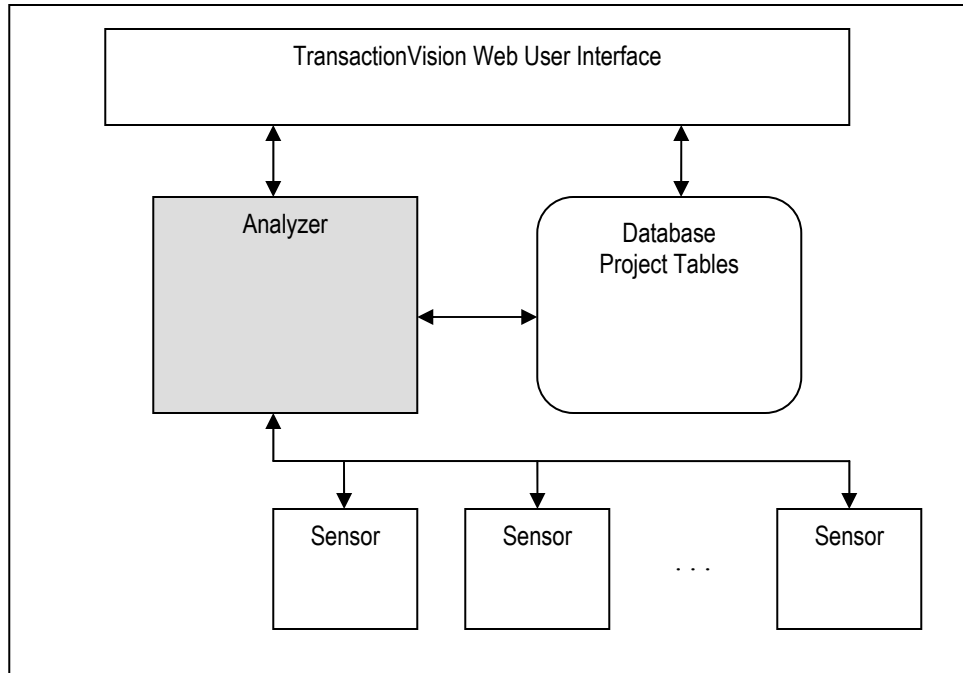
TransactionVision Analyzer Overview

The TransactionVision Analyzer is a service that communicates with TransactionVision Sensors via WebSphere MQ or JMS services. 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.

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.

Each TransactionVision project is assigned a single host running the Analyzer. Projects enable you to easily group and manipulate communication links, data collection filters, database schemas, and Analyzers as one entity. When you start a project, the Analyzer on the host assigned to the project is started automatically. You may also start the Analyzer on a host from the Analyzers page in the web user interface, in which case the Analyzer starts processing events on all active projects it is assigned to.

The following diagram shows the relationship of the TransactionVision Analyzer to other TransactionVision components:



Installation Overview

The TransactionVision installation consists of the following three TransactionVision packages:

- The Analyzer binaries and configuration files. Install this package on all hosts where you want the Analyzer service to run. This guide provides instructions for installing the Analyzer.
- The TransactionVision Web User Interface, which consists of WebSphere or WebLogic related files such as the .ear file, JSPs, servlets, etc. Install this package on the hosts that users and administrators will access via web browsers to use and manage TransactionVision. For instructions on installing and configuring the Web User Interface, see the *TransactionVision Web Application Installation and Configuration Guide*.
- TransactionVision Sensors. Install these packages on all hosts running applications that you wish to collect information about. For instructions on installing and configuring Sensors, see the *TransactionVision Sensor Installation and Configuration Guide*.

Installation steps vary for the different TransactionVision components on different platforms.

Before installing Analyzers, make sure the systems you are installing on meet the software requirements for TransactionVision. Requirements vary

for each package (Sensors, Analyzer, and Web User Interface). See the TransactionVision Release Notes for detailed software requirements.

Upgrading from Previous TransactionVision Releases

The Java Servlet, JMS and EJB Sensors from TransactionVision 4.2.1 can be used with the TransactionVision 5.0.0 Analyzer. TransactionVision 5.0.0 Sensors may not be used with older versions of the TransactionVision Analyzer.

Additional TransactionVision Resources

Documentation Roadmap

This guide provides instructions for installing and configuring the TransactionVision Analyzer. In addition to this guide, the following documents are provided with TransactionVision:

- The *TransactionVision Planning Guide* provides information to help you plan the TransactionVision implementation in your environment.
- The *TransactionVision Web Application Installation and Configuration Guide* provides instructions for installing and configuring the TransactionVision web user interface. This file is also available from the TransactionVision Help menu.
- The *TransactionVision Sensor Installation and Configuration Guide* provides instructions for installing and configuring TransactionVision Sensors. This file is also available from the TransactionVision Help menu.
- The *TransactionVision Administrator's Guide* provides instructions managing user accounts and communication links, configuring projects and data collection filters, and managing services and schemas. This file is also available from the TransactionVision Help menu.
- The *TransactionVision User's Guide* provides instructions using TransactionVision analysis views. This file is also available from the TransactionVision Help menu.
- The *TransactionVision Programmer's Guide* provides information for creating custom beans and reports for use with TransactionVision.
- The *TransactionVision Security Guide* provides an overview of the security features and setup procedures of TransactionVision. These features and procedures ensure that data collected by

TransactionVision is secure and accessible to the appropriate people.

Contacting Bristol Support

If you encounter a problem installing or configuring TransactionVision, contact Bristol Support by any of the following methods:

- TransactionVision: Choose the Help > Bristol Support menu item. From the submenu, you may open the Bristol Support home page, open the TransactionVision knowledgebase, or open a problem report form.
- Web: <http://www.bristol.com/support>
- Email: support@bristol.com
- Phone: **203-798-1007** Press 3

Chapter 2

Installing the Analyzer on Distributed Platforms

To install the TransactionVision Analyzer on distributed (UNIX) platforms, perform the following steps.

1. Change to the directory location of the TransactionVision installation files (either a CD device or download directory).
NOTE: On Solaris and HP-UX, you must copy the installation files from CD device to a temporary directory on your host's hard drive.

2. Login as superuser:

```
su
```

3. Enter the following command to begin the installation procedure:

```
./install.sh
```

The following menu is displayed:

```
This script will install/uninstall different  
TransactionVision components.
```

```
Unzipping/Untaring common package files ...  
Unzipping/Untaring Analyzer package files ...  
Unzipping/Untaring web package files ...  
Unzipping/Untaring Application Server sensor package  
files ...  
Unzipping/Untaring JMS sensor package files ...  
Unzipping/Untaring WebSphere MQ sensor package files  
...
```

Important! The six "Unzipping..." lines above are only for Solaris and HP-UX installations.

The following TransactionVision packages are available for installation:

1. TransactionVision Analyzer
2. TransactionVision Web
3. TransactionVision Application Server Sensor
4. TransactionVision JMS Sensor

5. TransactionVision WebSphere MQ Sensor

99. All of above

q. Quit install

Please specify your choices (separated by ,) by number/letter:

Important! Note that actual options and numbers depend on the installation files available on your computer: Also note that the Application Server Sensor package installs both the Servlet and EJB Sensors.

If this is the first time installing TransactionVision components on this computer, continue with “Initial Installation.” If an earlier version of TransactionVision is installed on this computer, continue with “Upgrade Installation.”

Initial Installation

1. To install only a single component, type the number associated with the TransactionVision component package and press Return. In this example, type **1** and press Return to install only the Analyzer.

To install multiple, but not all components, type the numbers associated with the components you wish to install, separated by commas, and press Return.

To install all available components, type **99** and press Return.

The installation script installs the specified package(s), then displays the menu again.

2. To quit the installation procedure, type **q** and press Return. To install additional components, see the installation instructions for those components.

Upgrade Installation

1. To install only a single component, type the number associated with the TransactionVision component package and press Return. In this example, type **2** and press Return to install only the Web Application.

To install multiple, but not all components, type the numbers associated with the components you wish to install, separated by commas, and press Return.

To install all available components, type **99** and press Return. See the installation instructions for the other components you wish to install for information about installing those components.

If the installation script determines that a previous version of TransactionVision is installed, it displays the following message:

```
There is an earlier version of TransactionVision
installed on the system.
The earlier version has to be uninstalled before
installing the current package(s).
Continue with the uninstallation? (Y/N) [N]:
```

2. Type **Y** and press Return to uninstall the previous version. If you type **N** and press Return, the installation process ends without uninstalling the previous version or installing the new version.

Before uninstalling the previous version, TransactionVision provides the option of migrating configuration files and project database schemas to the new installation:

```
Installation has detected an previous installation of
TransactionVision.
Migration converts configuration files and database
table schemas from the previous installation to the
new installation.
This step is required for TransactionVision to work
correctly with existing projects generated by an
earlier version of TransactionVision.
Continue to migrate TransactionVision project tables
and backed up configuration files from the previous
version? (Y/N) [Y]
Answer N and the database migration step can be
performed at a later time using the MigrateDB.sh
script.
```

3. Type **Y** and press Return to migrate configuration files and project database schemas from the previous installation to the new version. The installation script automatically creates a backup copy of existing configuration files for the migration. It then uninstalls the previous version of TransactionVision, installs the new Sensor package (and addition components, if specified), and displays the installation menu. Continue to step 4.

If you choose not to migrate configuration files and project database schemas at this time, you may migrate project database schemas at a later time, after configuring TransactionVision

components with the **MigrateDB.sh** script. See Appendix A, “Utilities Reference,” for information about this utility.

If you type **N** and press Return, the installation script provides the option of making a backup copy of configuration files for future reference:

```
Back up the configuration files from the previous
installation? (Y/N) [Y]
  Answer N will over write those files, causing any
  setup information to be lost.
```

Type **Y** and press Return to create a backup copy of your configuration files. The installation script prompts you to specify a backup location:

```
Backup copy of configuration files to
[/opt/TVision/migrate_tv421_date_time]:
```

Press Return to use the default location, or enter the desired backup directory location. The installation utility performs the following tasks:

- Copies the current configuration files to the specified directory
 - Uninstalls the previous version of TransactionVision.
 - Installs the new web package (and additional components, if specified).
 - Displays the installation menu again.
4. To quit the installation procedure, type **q** and press Return.

Installation Files

The following table shows the installation file names for the TransactionVision web package for each distributed platform.

Platform	Files
AIX	tvision_common_500_aix.bff tvision_analyzer_500_aix.bff
Linux	tvision_common_500_linux.rpm tvision_analyzer_500_linux.rpm
Solaris	tvision_common_500_solaris_tar.gz tvision_analyzer_500_solaris_tar.gz

Uninstalling the Analyzer

To uninstall TransactionVision components, perform the following steps:

1. Login as superuser:

```
su
```

2. Enter the following command:

```
install.sh -u
```

The following menu is displayed (note that actual options depend on the TransactionVision packages installed on your computer):

```
The following TransactionVision packages are
installed on the system:
```

- ```
1. TransactionVision Web
2. TransactionVision Analyzer
3. TransactionVision Application Server Sensor
4. TransactionVision JMS Sensor
5. TransactionVision WebSphere MQ Sensor
```

- ```
99. All of above
q. Quit uninstall
```

```
Please specify your choices (separated by ,) by
number/letter:
```

3. Type the number associated with the TransactionVision package you wish to uninstall (**2** for the Analyzer in this example) and press Return.

To uninstall all TransactionVision components, type **99** and press Return.

The installation script uninstalls the specified package, then displays the menu again.

If you uninstall the servlet Sensor, it will be first turned off in the WebSphere Application Server the Sensor is monitoring. The instrumented jar files under \$WAS_HOME/classes will be removed.

4. To quit the uninstall, type **q** and press Return. If the common package is the only TransactionVision package still installed, it will be uninstalled automatically.

Important! After uninstalling the TransactionVision web user interface, you must clean up its temporary cache and distribution directory. WebSphere does not do this automatically, and any old files could cause a new installation to work incorrectly.

Chapter 3

Installing the Analyzer on Windows

The TransactionVision Analyzer and web user interface are installed as a single package on Windows. Note that you must be logged into the target system either as Administrator or as a user with Administrator privileges.

For migration to work properly on Windows when you are upgrading from a previous release, either the `JAVA_HOME` environment variable must be set or Java must be present in your path before you begin the installation. Migration requires Java; if neither of these is set, the installation may fail to migrate configuration files and you may need to run `TVisionSetupInfo` manually.

To install this package, perform the following steps:

1. Close all Windows programs currently running on your computer.
2. In the Windows Explorer, double-click `tvision_500_win.exe`. The InstallShield Welcome screen appears.
3. Click `Next>` and wait until the TransactionVision Setup Welcome screen appears.

If this is the first time installing the TransactionVision Web Application on this computer, continue with “Initial Installation.” If an earlier version of the TransactionVision Web Application is installed on this computer, continue with “Upgrade Installation.”

Note: If the InstallShield Save Files screen appears, click `Next>` to use the default folder for extracting installation files (for example, `C:\TEMP\Bristol TransactionVision`), or click `Change` to select the desired folder and click `Next>` to continue.

Initial Installation

For an initial installation, the Setup Welcome screen is displayed.

1. On the Setup Welcome screen, click Next> to display the TransactionVision license agreement.
2. Click Yes to accept the license agreement. The User Information screen appears.
3. Enter your name and company name, then click Next>. The Destination Location screen appears.
4. To use the default installation folder (C:\Program Files\Bristol TransactionVision), click Next>. To choose a different installation folder, click Browse..., select the desired installation folder, then click Next>. The Setup Type screen appears.
5. To install both the Analyzer and web user interface, select Complete and click Next>. To install only the Analyzer, select Custom, click Next>, deselect the web user interface, and click Next>.

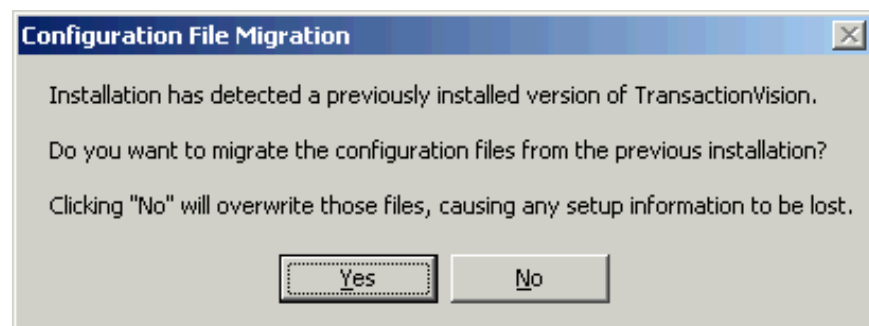
The selected packages are installed in the specified location. The Setup Complete page appears.

6. Click Finish to complete the installation.
7. If you are running WebLogic as a Windows service, continue with the steps in “Modifying the Registry for WebLogic.”

Upgrade Installation

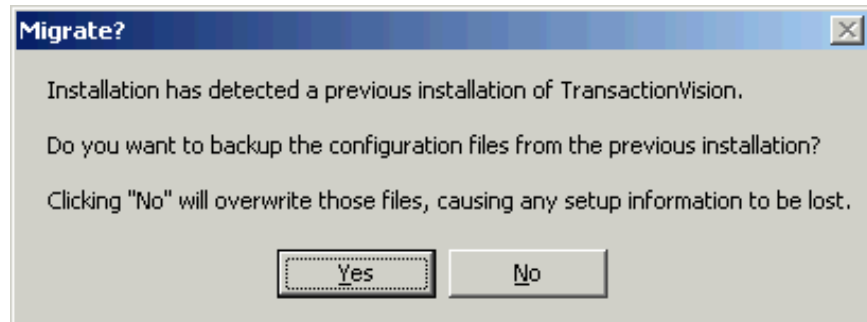
For an upgrade installation, the installation wizard displays the setup maintenance menu.

1. If you wish to install TransactionVision with **different** settings from the previous installation, select Remove and click Next> to uninstall the previous installation, then begin the installation procedure again. If you are upgrading from a previous release, select Reinstall and click Next> to install TransactionVision using the settings from the previous installation. The Configuration File Migration dialog appears:



2. To maintain configuration information from the previous installation, click Yes. The installation wizard makes a backup copy of existing configuration files, installs the new version of TransactionVision, and opens an MS-DOS window to migrate existing configuration files to the new version. When complete, the Setup Complete screen appears.

To overwrite existing configuration files, click No. The installation wizard displays a message box asking whether you want to make a backup copy of existing configuration files before continuing the installation.



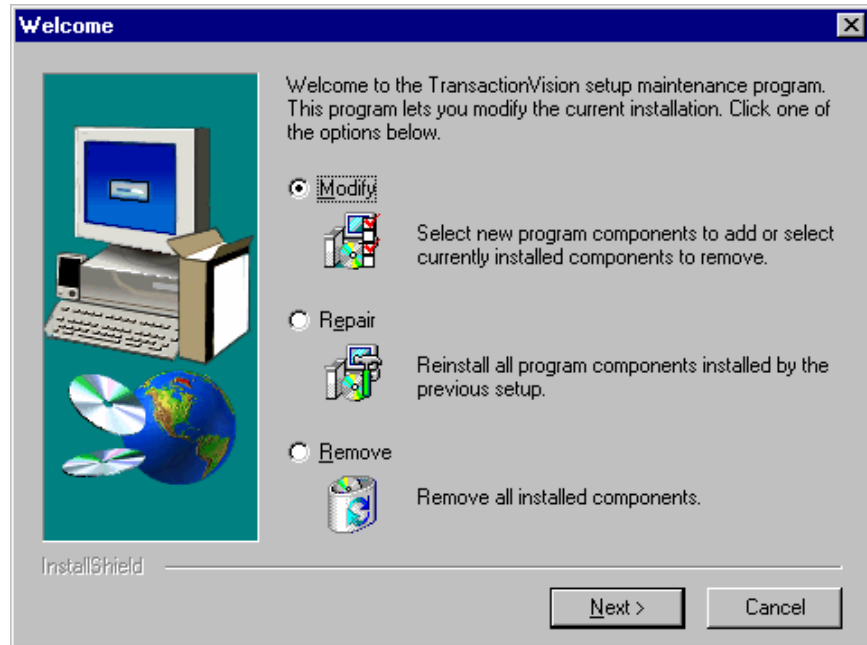
Click Yes to create a backup copy or No to continue the installation without backing up configuration files. The installation wizard then installs the new version of TransactionVision, overwriting existing configuration files, and displays the Setup Complete screen.

3. Click Finish to complete the installation.

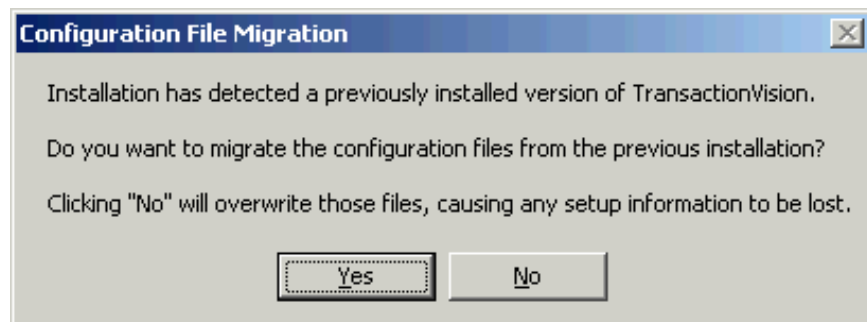
Modifying the Installation

After you install TransactionVision Sensors on a host, you may wish to modify your installation. For example, suppose you initially install the Analyzer on a host, then later decide to install the Servlet and JMS Sensors. To modify your installation perform the following steps:

1. Close all Windows programs currently running on your computer.
2. In the Windows Explorer, double-click `tvision_500_win.exe`. The InstallShield Welcome screen appears.
3. Click Next> and wait until the TransactionVision Setup Welcome screen appears.



4. To install an additional component or remove an installed component, select Modify. To reinstall all TransactionVision components installed by the previous setup, select Repair. To uninstall all components, select Remove.
5. Click Next> and select the additional components you wish to install. The Configuration File Migration dialog appears:



6. Click No, because migration is not required for installing additional components. The installation wizard displays a message box asking whether you want to make a backup copy of existing configuration files before continuing the installation. Click No again to continue the installation without backing up configuration files. The installation wizard then installs the selected TransactionVision components and displays the Setup Complete screen.
7. Click Finish to complete the installation. If you wish to display the migration log file created when migrating configuration and

project database files to the new version, check View Migration Log File before clicking Finish.

Uninstalling the Analyzer

To uninstall TransactionVision components, perform the following steps:

1. From the Start menu, choose Settings > Control Panel.
2. Double-click Add/Remove Programs.
3. Select the Bristol TransactionVision package you wish to uninstall and click Change/Remove. The maintenance menu screen appears.
4. Select Remove and click Next> to remove TransactionVision components.
5. Click OK to confirm that you wish to uninstall the specified package. The specified package is uninstalled. The following types of files are **not** deleted:
 - Any files added after the installation
 - Any shared files associated with packages that are still installed

If shared files do not appear to be associated with any installed packages (for example, if all other TransactionVision packages have been uninstalled), the Shared File Detected screen appears.

- To leave all shared files installed, check Don't display this message again and click No.
- To leave the current file, but display this message for any other shared files, click No.
- To delete the shared file, click Yes.

If you uninstall the servlet Sensor, it will be first turned off in the WebSphere Application Server the Sensor is monitoring. The instrumented classes under `$WAS_HOME/classes` will be removed along with its parent directories if they are empty.

6. The Uninstallation Complete screen appears. Click Finish to complete the uninstallation procedure.

Important! After uninstalling the TransactionVision web user interface, you must clean up its temporary cache and distribution directory. WebSphere does not do this automatically, and any old files could cause a new installation to work incorrectly.

Chapter 4

Configuring Databases

Before configuring the TransactionVision Analyzer, it is important to ensure that your DBMS environment is configured properly to work with TransactionVision. This configuration differs for DB2 and Oracle databases.

It is first necessary to create your DB2 or Oracle database before running the `TVisionSetup` script. The `TVisionSetup` script does **not** create a database and relies on a connection to an existing database. Database variables or tuning parameters must then be adjusted to ensure that the database will operate at an acceptable level of performance for your application. Several tools are provided with TransactionVision to aid in the identification and resolution of database performance bottlenecks.

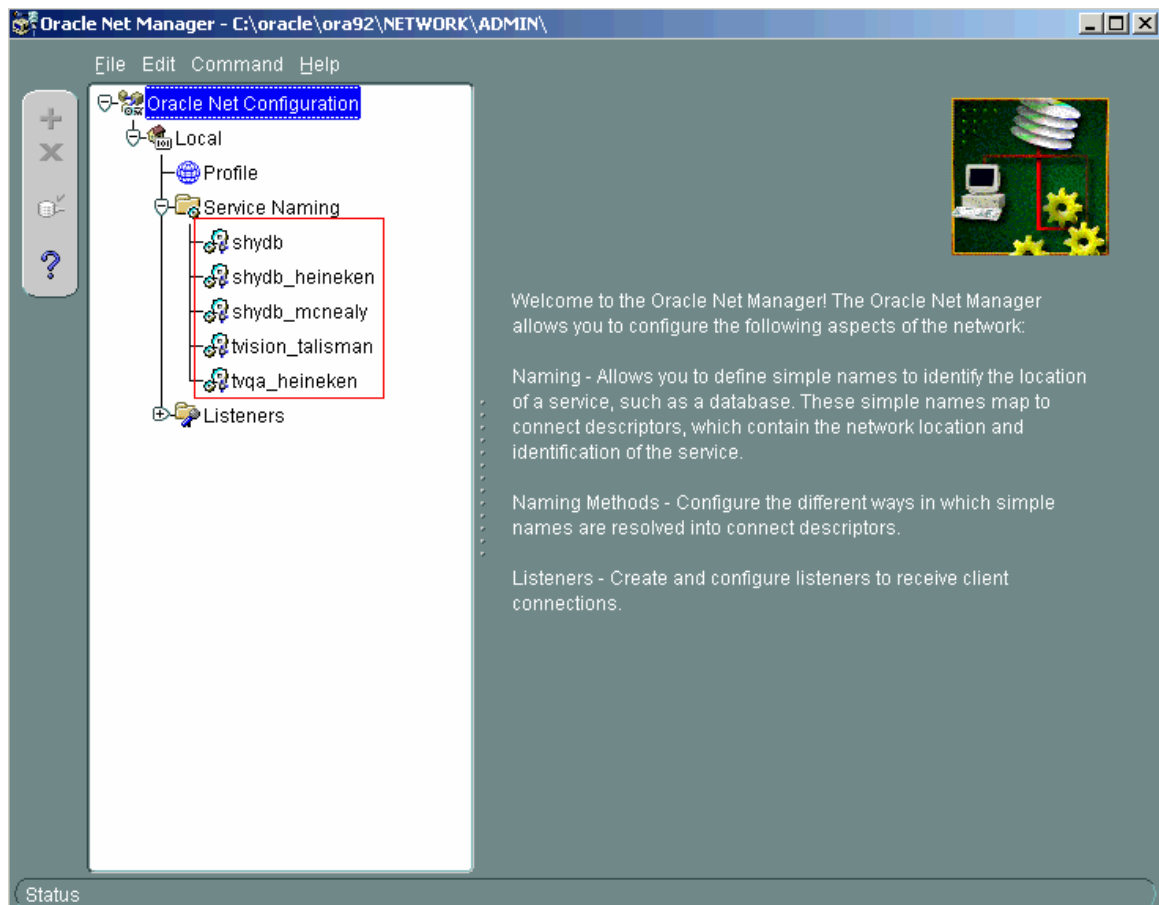
Identifying Database Connection Names

When you run the `TVisionSetupInfo` utility to configure the TransactionVision Analyzer and Web Application, you will be prompted for the database connection name. This name depends on which database provider you are using and, for Oracle, which database client type.

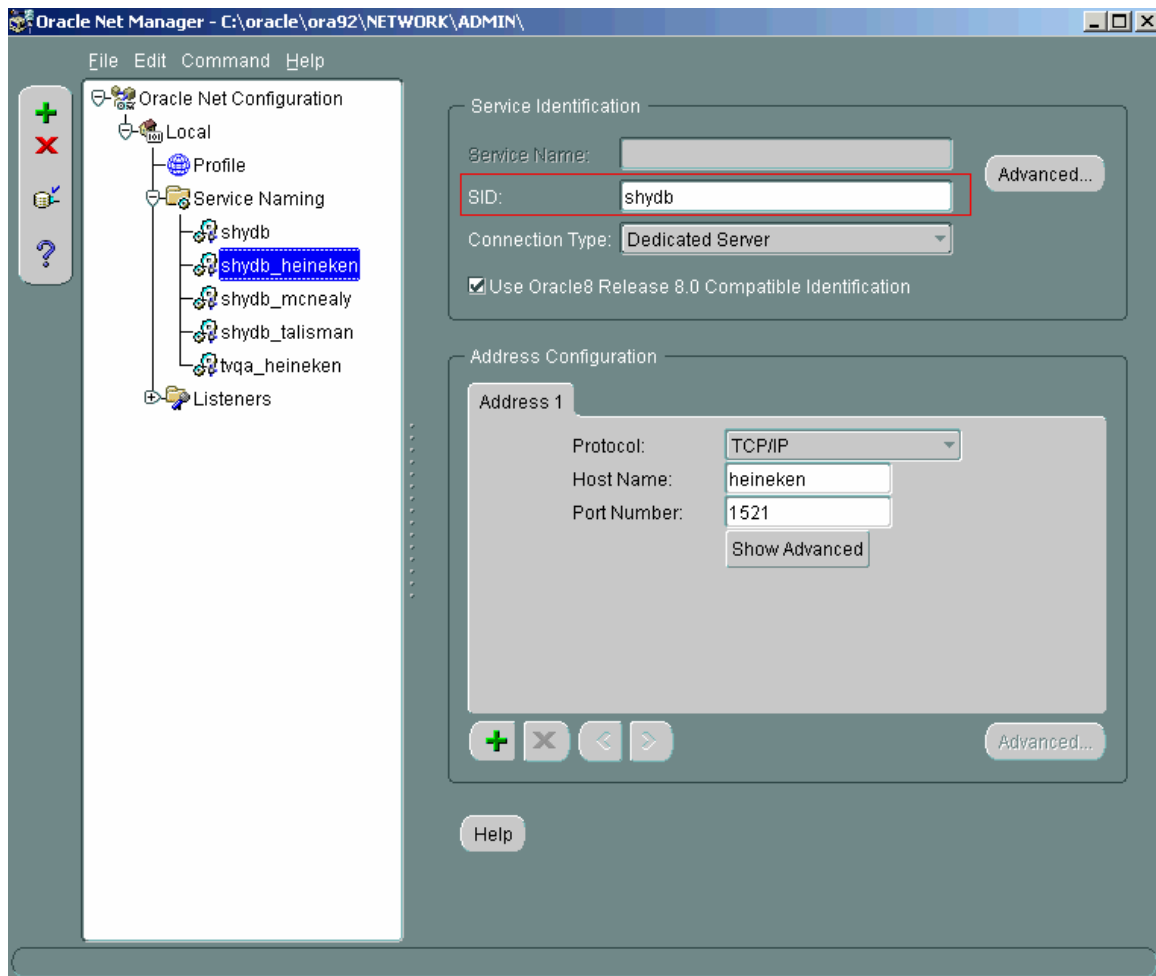
Oracle OCI Client

If you are using an Oracle oci client with TransactionVision, you will be prompted to enter the database connection name and database name.

The connection name is the Oracle Net Service Name. Use the Oracle Net Manager to find all net service names. Choose the appropriate connection name from the Service Naming folder (for example, `TVQA_HEINEKIN`).



The database name is the Oracle System Identifier (SID) (for example, tvqa). To identify the SID, click the service name. In the following example, shydb is the database name for the shydb_heineken service name.



Oracle Thin Client

If you are using the Oracle thin driver with TransactionVision, you will be prompted to enter only the database connection name. The connection name is the Oracle SID (for example, *tvqa*).

IBM DB2

If you are using DB2, you will be prompted to enter the database connection name and the database name. The connection name is the DB2 database alias name. To find the alias name, use the following command:

```
db2 list database directory
```

For local databases, the alias name is typically the same as the database name, unless a remote loopback is defined.

For remote databases, the alias name will typically be different from the database name.

In the following example, the database connection name is TVDB and the database name is TVISION:

```
C:\Program Files\IBM\WebSphere MQ\Java\bin>db2 list
database directory
System Database Directory

Number of entries in the directory = 13

Database 1 entry:

Database alias                = TVDB
Database name                 = TVISION
Node name                    = TCP7768C
Database release level       = a.00
Comment                      =
Directory entry type         = Remote
Catalog database partition number = -1

Database 2 entry:

Database alias                = BACARDI
Database name                 = TVISION
Node name                    = NDE682FE
Database release level       = a.00
Comment                      =
Directory entry type         = Remote
Catalog database partition number = -1
```

Setting DB2 Variables

Before using TransactionVision, set the values for the following DB2 variables to the values shown in the following table:

Variable	Value	Description
APP CTL HEAP SZ	1024	Maximum application control heap size. This value indicates the number of 4KB blocks.
MAXAPPLS	150	Maximum number of active applications
APPLHEAPSZ	1024	Default application heap size. this value indicates the number of 4KB blocks.
LOCKLIST	Estimate based on system	Amount of storage (in 4KB units) allocated to the lock list.

In addition, you should increase the size of the bufferpool.

APP CTL HEAP SZ, MAXAPPLS, and APPLHEAPSZ

Use the following commands to set these values. Note that the last three commands will drop all active database connections and then stop and start the DB2 server. Be sure to run these steps at an appropriate time when other database users will not be affected.

```
db2 connect to tvision
```



```
db2 get db cfg for tvision
db2 update db cfg for tvision using APP_CTL_HEAP_SZ 1024
db2 update db cfg for tvision using MAXAPPLS 150
db2 update db cfg for tvision using APPLHEAPSZ 1024
db2 force application all
db2stop
db2start
```

In the environment in which WebSphere is started on UNIX systems, the DB2INSTANCE environment variable must also be set to the DB2 instance being used by TransactionVision.

LOCKLIST

To determine the number of pages required for your lock list, perform the following steps:

1. Calculate a lower bound for the size of your lock list:

$$(512 * 36 * \text{mapappls}) / 4096$$

where 512 is an estimate of the average number of locks per application and 36 is the number of bytes required for each lock against an object that has an existing lock.

2. Calculate an upper bound for the size of your lock list:

$$(512 * 72 * \text{maxappls}) / 4096$$

where 72 is the number of bytes required for the first lock against an object.

3. Estimate the amount of concurrency you will have against your data. Based on your expectations, choose an initial value for LOCKLIST that falls between the upper and lower bounds that you have calculated. Use the following command to set the value:

```
db2 update db cfg for database_name using LOCKLIST n
```

where *n* is the value for LOCKLIST.

4. Using the database system monitor, tune the value of this parameter. For more information, see Chapter 32, “Configuring DB2 Capacity Management”, in the *DB2 V7 Administration Guide*.

DB2_RR_TO_RS

Multiple threads performing concurrent database operations along with a DB2 behavior called Next Row Locking may lead to database deadlocks. If you plan to use multiple event collection threads, set the DB2 variable DB2_RR_TO_RS to ON to avoid deadlocks:

```
db2set DB2_RR_TO_RS=ON
```

This setting is only effective after a DB2 restart. It affects all DB2 applications and cannot be used if other non-TransactionVision applications that require “Repeatable Read” semantics (Transaction Isolation Level RR) are using the same DB2 instance. If other applications using the same instance require “Repeatable Read” semantics, you must either create a separate DB2 instance for TransactionVision or set the number of event collection threads to 1. For information on setting the number of event collection threads, see the “Managing Communication Links” chapter in the *TransactionVision Administrator’s Guide*.

Bufferpool

The size of the default DB2 bufferpool is only 250 pages. Increase it to 1024 (or more if sufficient memory for DB2 is available) to improve performance. The following command will change the size for the default bufferpool from 250 pages to 1024 pages:

```
db2 ALTER BUFFERPOOL IBMDEFAULTBP SIZE 1024
```

Setting Oracle Variables

If it is expected that TransactionVision will be used in a relatively simple environment where minimal database connections are required, no special configuration is required for the Oracle DBMS. However, environments where a large number of users will be simultaneously accessing the web user interface, several Analyzers will be in use, or where the Analyzer will incorporate higher than the default number of threads, it will be necessary to increase the Open Cursors database parameter. Related error messages may be expected to show up in the TransactionVision UI or Analyzer logs if this limit is exceeded. To increase the number of Open Cursors, execute the following command:

```
alter system set open_cursors = 600
```

This change may be made dynamically while the Oracle server is running. It is not necessary to restart the RDBMS.

Connecting to an Oracle Database

To connect to an Oracle database, perform the following steps:

1. If you are planning to use the Oracle oci driver to make database connections, make sure the target database is defined in your `tnsnames.ora` file. For example:

```
TESTDB_SIMPLEX6.BRISTOL.COM =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST =
```

```
simplex6) (PORT = 1521))
)
(CONNECT_DATA =
(SID = testdb)
(SERVER = DEDICATED)
)
)
```

Various Oracle tools can create such an entry for you, or you can add it manually.

2. Run the `rvisionSetupInfo` script (see Chapter 5, "Configuring the Analyzer," for instructions) and enter Oracle when you are prompted to enter the type of database to use, enter desired client type (either `oci` or thin clients). For `oci`, enter your database name for database connection name and SID for database name. For thin clients, enter the database listener port and SID for database connection name.

This script updates the `Database.properties` file based on the settings you select. For example, for an `oci` client connection, the `Database.properties` file will contain entries similar to the following example:

```
jdbc_driver=oracle.jdbc.driver.OracleDriver
database_connection_name=TESTDB_SIMPLEX6
database_name=testdb
database_host=simplex6
database_port=1521
oracle_client_type=oci
```

For a thin client connection, the `Database.properties` file will contain entries similar to the following example:

```
jdbc_driver=oracle.jdbc.driver.OracleDriver
database_connection_name=testdb
database_name=testdb
database_host=simplex6
database_port=1521
oracle_client_type=thin
```

For more information about these entries, see "Database Properties" in Chapters 4 and 5.

DBMS Performance 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.

Optimizing I/O Throughput

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.

Prior to deployment, it is imperative 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 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. DB2 log files and Oracle Rollback Segments hold uncommitted database operations and usually are highly utilized during database insert/update. For this reason they should have their own containers 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.

There are many other database parameters that may impact the performance of TransactionVision. For DB2 in particular, those parameters previously mentioned in “Setting DB2 Variables” must be examined one-by-one to ensure they are optimized.

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

Testing DBMS and Diagnosing Performance Bottlenecks

Bristol provides independent tools, **DB2Test** and **OracleTest**, 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 where the TransactionVision Analyzer will be installed. For more information about these tools, see Appendix A, “Utilities Reference.”

The tools simulate the database update operation generated by TransactionVision. Run the test multiple times to get a complete picture of the DBMS performance. Please 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.

- Run the insert test with one thread and with record size of 1KB - this will gauge the raw event insert performance.
- 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.
- Run the insert test with one thread and with record size of (13KB + average message size) - this will gauge the analyzed event insert performance.
- Run the insert test with multiple threads and with record size of (13KB + average message size). 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.
- 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 Bristol. During the test the following parameters of the DBMS system should be monitored:

- Disk I/O usage for all involved physical disks (tablespaces and log files), especially I/O busy percentage.
- 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 database, DBMS and kernel parameters with Bristol for any configuration issues.

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

Updating Statistics with RUNSTATS

The database RUNSTATS command updates statistics about the physical characteristics of a table and the associated indexes. These characteristics include number of records, number of pages, and average record length. The optimizer uses these statistics when determining access paths to the data.

This command is called when a table has had many updates, such as when data is continuously collected into DB2 by the TransactionVision Analyzer. It could result in large performance gains in queries made by TransactionVision views and reports, as well as queries made internally by the TransactionVision Analyzer to correlate events.

TransactionVision provides the following scripts for running the RUNSTATS command on TransactionVision tables:

- `DB2RunStats.[sh|bat]` for a DB2 environment
- `OracleRunStats.[sh|bat]` for an Oracle environment

Important! Oracle 10g has a built-in job scheduler that automatically collects statistics. You do not need to run this script for version 10g.

These scripts can be set up to run as scheduled batch jobs using either the UNIX `crontab` facility or the Windows scheduler.

Important! While these scripts are running, TransactionVision Analyzer processing slows down.

These scripts typically should be run daily, though the frequency of execution could be higher for higher message rates.

Run these scripts from a user account that has privileges to perform database operations. The usage is as follows:

```
DB2RunStats database_name schema_name [-v7]
OracleRunStats username passwd database_name schema_name
```

The `database_name` option represents the name of the database to connect to, and `schema_name` is the name of the schema that the project reads and writes data to. For `DB2RunStats`, use the `-v7` flag for DB2 7.x; the default operation is for DB2 8.1.

Important! These scripts need to be customized based on your system to invoke the right environment initialization script like `.profile` or `.bashrc`, and to set the correct DB2 installation location and the correct `TVISION_HOME` location. For `OracleRunStats`, the file `OracleRunStats.sql` **must** be in the current directory before the script is executed.

Example cron Job

For a UNIX environment, the following example is a typical crontab entry for executing the **DB2RunStats** script every night:

```
5 0 * * * <TVISION_HOME>/bin/DB2RunStats.sh database-name  
schema-name > <TVISION_HOME/logs/runstats.log 2>&1
```

In this example, `<TVISION_HOME>` is the TransactionVision installation directory, `database-name` is the database name to connect to, and `schema-name` is the schema on the tables to run RUNSTATS on.

The above entry executes the script every night 5 minutes after midnight. A crontab entry needs to be added for every project schema being used by TransactionVision. Use `crontab -e` to create a cron job entry. For details, refer to the man pages of cron and crontab.

Example Scheduled Task

For a Windows environment, run these scripts through Windows scheduler at regular intervals. Add a scheduler entry for every project schema being used by TransactionVision. The Windows scheduler can be accessed through the Windows Control Panel. Make sure this script runs in the DB2 command window.

The following is an example command for the Windows scheduler to run the **DB2RunStats** script:

```
"<DB2-install-dir>\bin\db2cmd.exe" "<TVISION_HOME>\bin\  
DB2RunStats.bat" database-name schema-name > "<TVISION_  
HOME>\logs\runstats.log"
```

In this example, `<TVISION_HOME>` is the TransactionVision installation directory, `database-name` is the database name to connect to, and `schema-name` is the schema on the tables to run RUNSTATS on.

DBMS Disk Space Requirements

The other factor that needs to be finalized is the amount of disk storage space required for TransactionVision events. This is determined by the average size of the 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:

(Average message size + 13K Byte) x Event Rate x Event Retention Time

For example, if the average message size is 2K Bytes, 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 ((2 + 13)K Bytes x 720,000 x 2 x 4 = 90G Bytes).

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

Configuring Databases for Unicode Data

TransactionVision can display Unicode data in views and reports. However, you must create the TransactionVision database with the required code/character set, and set the appropriate database property within TransactionVision.

Database Code/Character Set

When you create the TransactionVision database, you must specify the properties shown in the following table:

Database Provider	Required Settings
DB2	The TransactionVision database must be created with Code Set UTF-8.
Oracle	<ul style="list-style-type: none">The TransactionVision database must be created with the character set AL32UTF8 or UTF8.The NLS_LENGTH_SEMANTICS initialization parameter must be set to BYTE.

TransactionVision Database Properties

In the `Database.properties` file, set the `unicode_db` property to `true`. This setting affects the size of certain columns when project schemas are created through `CreatesQLScript` or the TransactionVision graphical user interface. The size attribute of character type XDM columns specifies the number of characters that can be stored for the corresponding table column. If the `unicode_db` property is set, all character-based XDM columns with the attribute `unicode=true` set will be generated with double the byte size to allow the specified number of characters to be stored in the database. For more information on modifying the `Database.properties` files, see Appendix B.

Chapter 5

Configuring the Analyzer

Configuration Overview

Once TransactionVision is installed, you must create an IBM DB2 or Oracle database for TransactionVision to save event data. Refer to Chapter 4, "Configuring Databases," for the required database parameters. The TransactionVision setup utilities do not create the database.

After the database is created, the following setup tasks must be completed before you can use TransactionVision Analyzer:

- Create the TransactionVision database schema.
- Start the Analyzer service.

TransactionVision provides two utilities to guide you through the setup process:

- The `TVisionSetupInfo` utility enables you to enter information it needs for the setup process and sets required environment variables.
- The `TVisionSetup` utility uses information you provide to `TVisionSetupInfo` to perform setup tasks.

Important! This chapter provides instructions for configuring the Analyzer only. If you are installing both the Analyzer and Web Application on a single system, refer to the *TransactionVision Web Application Installation and Configuration Guide* for instructions on installing the Web Application and then configuring both components.

Using `TVisionSetupInfo`

If only the Analyzer is installed, the `TVisionSetupInfo` utility modifies the following files:

- `TVISION_HOME/config/datamgr/Database.properties`
- `TVISION_HOME/config/setup/Setup.properties`

In addition, `TVisionSetupInfo` does the following:

- Saves the installation path for software tools in `TVISION_HOME/config/setup/DefaultInstallPath.xml`
- Generates `TVISION_HOME/bin/SetupEnv.[sh|bat]`, which is run by `TVisionSetup` to set the `JAVA_HOME`, `CLASSPATH`, and system library path environment variables required by TransactionVision

Required Information

The `TVisionSetupInfo` utility prompts you to enter the following information necessary to complete the configuration. Please review these information requirements before running `TVisionSetupInfo` so you will be prepared to enter appropriate responses.

The following table shows parameters and associated technology areas that will be required. See the following sections for detailed descriptions of the parameters.

Parameter	Technology			
	WebSphere	TIBCO	DB2	Oracle
Log file location				
DB2INSTANCE			✓	
Database connection name			✓	✓
Database name			✓	✓
Database host			✓	✓
Database user name			✓	✓
Database user password			✓	✓
Database driver				✓
Database listener port				✓

Parameter	Technology			
	WebSphere	TIBCO	DB2	Oracle
Sensor types				
Message system provider				
Database installation path			✓	✓
WebSphere MQ installation path	✓			
TIBCO EMS installation path		✓		
Additional JAR files				

Log File Location

You must supply the pathname of the directory where TransactionVision should store log files. The default value is `TVISION_HOME/logs`.

Database Properties

You must specify the type of database you plan to use with TransactionVision (IBM DB2 or Oracle).

For a DB2 database, you must supply the following information:

- The value of the DB2INSTANCE environment variable. The default value is DB2. Before starting your WebSphere 5 server, your DB2INSTANCE environment variable **must** be set. Failure to set this variable will cause a failure during TransactionVision startup and cause the WebSphere server to fail to start and initialize.
- The name of the database connection to be used by TransactionVision. See Chapter 4 for more information on identifying the connection name and database name.
- The name of the database that TransactionVision will connect to. This name may be different from the "database_connection_name" if a client database connection is used. See Chapter 4 for more information on identifying the connection name and database name.
- The name of the host that the database runs on. The default value is `local_host`.

- The user name for connecting to the database. If this field is empty, the currently logged in user is used to make the database connection. Make sure that the user specified or currently logged in has privileges for database access.
- The password for the database user name. If this field is empty, the currently logged in user's password is used to make the database connection. If **TVisionSetupInfo** is able to detect an installed JCE provider that supports the DES encryption algorithm, it sets the `jce_provider` value in the `Database.properties` file to the class name of the JCE provider and encrypts the password. Otherwise, it disables the password encryption feature and stores the password as unencrypted text.

For an Oracle database, you must supply the following information:

- The driver to use. For Oracle 9, you may specify `thin` or `oci`; `thin` is the recommended client type. **Note:** For IBM WebSphere or BEA WebLogic Application Servers to use Oracle JDBC OCI driver, `ORACLE_HOME` must be set in the user profile. Also make sure that the Oracle library is included in the dynamic library load path.
- The port number for the Oracle listener.
- The name of the database connection to be used by TransactionVision. The default value is `TVISION`. See Chapter 4 for more information on identifying the connection name and database name.
- The name of the database that TransactionVision will connect to. The default value is `TVISION`. This name may be different from the "database_connection_name" if a client database connection is used. See Chapter 4 for more information on identifying the connection name and database name.
- The name of the host that the database runs on. The default value is `local_host`.
- The user name for connecting to the database. If this field is empty, the currently logged in user is used to make the database connection. Make sure that the user specified or currently logged in has privileges for database access.
- The password for the database user name. If this field is empty, the currently logged in user's password is used to make the database connection. If **TVisionSetupInfo** is able to detect an installed JCE provider that supports the DES encryption algorithm, it sets the `jce_provider` value in the `Database.properties` file to the class name of the JCE provider and encrypts the password.

Otherwise, it disables the password encryption feature and stores the password as unencrypted text.

Messaging System Provider

You must specify the messaging system provider used to transfer TransactionVision configuration messages and events between Sensors and the Analyzer. Supported providers include WebSphere MQ and TIBCO EMS. You may specify one or both providers.

Installation Paths

You will be prompted to enter installation paths for the following required software:

- Database
- WebSphere MQ

Analyzer Settings

You must specify the following Analyzer information:

- The types of Sensors you are using in your environment
- The location of any JAR files that need to be added to the Analyzer CLASSPATH. These JAR files contain plugin beans that customize the Analyzer.

TVisionSetupInfo Syntax

1. Set the `TVISION_HOME` environment variable to the absolute path of the installation directory. For example, on Solaris, `TVISION_HOME` would be `/opt/TVision`; on AIX it would be `/usr/lpp/TVision`.
2. Run the `TVisionSetupInfo`. `[sh|bat]` script. This utility modifies TransactionVision configuration files, so you must have file modification privileges to run it (typically root for UNIX or Administrator for Windows).

Operating System Script Command

AIX, Linux, Solaris	<TVISION_HOME>/bin/TVisionSetupInfo.sh
Windows	<TVISION_HOME>\bin\TVisionSetupInfo.bat

When responding to user input prompts from `TVisionSetupInfo`, press Enter to accept the default value shown in brackets; otherwise, type the correct value and press Enter. To specify an empty value, press the spacebar, and then press Enter. Sample input and output below is shown in *italics*.

TVisionSetupInfo Prompts

Configuring Log Files

Checking Java version ...

This program collects configuration information in order to setup the TransactionVision environment. This includes:

- Location of software that TransactionVision depends on such as WebSphere MQ, the application server and the relational database system,
- Parameters required to connect to WebSphere MQ,
- Parameters to connect to the database system,
- Optional parameters to connect to the LDAP system
- Setup parameters for the TransactionVision Analyzer, and
- Setup parameters for the TransactionVision web component.

You will be prompted to input required configuration parameters. If a default value is provided in [], pressing <Enter> will set the parameter to this default value. Pressing <Space><Enter> will set the parameter to an empty value.

Please specify name of directory where you want to store your log files [C:\Program Files\Bristol\TVision\logs]:
C:\Program Files\Bristol\TransactionVision\logs

Modifying *.Logging.xml files in log file directory
C:\Program Files\Bristol\TVision\logs

```
0 [main] INFO AppLog - TransactionVision
Info(FileUpdated): File "C:\Program
Files\Bristol\TVision\config\logging\Analyzer.Logging.xml"
has been successfully updated
```

```
78 [main] INFO AppLog - TransactionVision
Info(FileUpdated): File "C:\Program
Files\Bristol\TVision\config\logging\Migrate.Logging.xml"
has been successfully updated
```

```
141 [main] INFO AppLog - TransactionVision
Info(FileUpdated): File "C:\Program
Files\Bristol\TVision\config\logging\Setup.Logging.xml" has
been successfully updated
```

```
219 [main] INFO AppLog - TransactionVision
Info(FileUpdated): File "C:\Program
Files\Bristol\TVision\config\logging\UI.Logging.xml" has
been successfully updated
```

Configuring Database Properties

Retrieving database configuration parameters ...

Type of Database (Oracle or DB2)? [DB2]: *DB2*

If you specify Oracle as the type of database, you will be prompted to specify the driver to use. For Oracle 9, you may specify `thin` or `oci`; `thin` is the recommended client type.

For a thin client type, you must also enter the port number that the Oracle listener is on. The default is 1521.

Value of your environment variable DB2INSTANCE [DB2]: *DB2*

Important! Before starting your WebSphere 5 server, your DB2INSTANCE environment variable must be set. Failure to set this variable will cause a failure during TransactionVision startup and cause the WebSphere server to fail to start and initialize.

Database connection name to be used by TransactionVision
[TVISION]: *TVDB*

For DB2, enter the name of the database connection to be used. For Oracle, enter the Oracle Local Net Service Name for an oci client connection or the SID (system identifier) for a thin client connection. See Chapter 4 for more information on identifying the connection name and database name.

Name of database TransactionVision connects to [TVISION]:
TVDB

For DB2, enter the name of the database on the server to connect to. For an Oracle oci client connection, enter the SID. This prompt does not appear for an Oracle thin client connection. See Chapter 4 for more information on identifying the connection name and database name.

Name of the host the database is running on [*local_host*]:
devpc

Enter the name of the host the database server is running on.

Database user name [db2admin]:

Enter the user name to be used while making the database connection. Leave blank to use the currently logged in user to make the database connection.

User password [ibmdb2]:

Enter the password to be used while making the database connection. Leave blank to use the currently logged in user's password to make the database connection.

The **TVisionSetupInfo** utility automatically searches for an installed JCE provider that supports the DES encryption algorithm. If such a provider is found, **TVisionSetupInfo** sets the `jce_provider` value in the `Database.properties` file to the class name of the JCE provider. If no JCE provider is found, **TVisionSetupInfo** displays the following message:

```
Java Cryptography Extension (JCE) is not present in your
current JDK. Password encryption feature will be disabled
and stored in plain text.
```

```
2004-08-23 18:14:40,338 [main] - TransactionVision
Info(FileUpdated): File C:\Program
Files\Bristol\TVision\config\datamgr\Database.properties"
has been successfully updated
```

Configuring Messaging System Provider

Please select your Queuing Product Type by number.

1. WebSphere MQ
2. TIBCO EMS
3. All of the above

Your Queuing Product [1]: 3

Enter the number corresponding to the messaging system provider you wish to use to transfer configuration and event messages between the Analyzer and Sensors. **TVisionSetupInfo** uses this information to generate the proper shell script to set up the Analyzer runtime environment correctly. If the Analyzer will collect event from both IBM WebSphere MQ communication links and TIBCO EMS communication links, select option 3.

Configuring Installation Paths for Dependent Software

```
Retrieving installation path for dependent software tools
...
```

```
DB2 installation location [C:\Program Files\sqllib]:
```

The following prompt only appears if you selected WebSphere MQ as a message system provider.

```
WebSphere MQ installation location [C:\Program
Files\IBM\WebSphere MQ]:
```


The following prompt only appears if you selected TIBCO EMS as a message system provider.

```
TIBCO EMS installation location [C:\tibco\ems]:  
  
2004-08-23 18:14:45,995 [main] - TransactionVision  
Info(FileUpdated): File C:\Program  
Files\Bristol\TVision\config\setup\DefaultInstallPath.xml"  
has been successfully updated
```

TransactionVision Analyzer Settings

Please select your Sensor types by number. This enables the Analyzer to receive events from the selected Sensors. If you are selecting multiple Sensor types, please separate them with ",".

1. CICS
2. EJB
3. WebSphere MQ JMS
4. TIBCO EMS JMS
5. Servlet
6. WebSphere MQ
7. WebSphere MQ IMS Bridge
8. All of the above

Your Sensor type(s) [8] :

Enter the number corresponding to the types of TransactionVision Sensors you plan to use. For multiple Sensor types, separate the numbers with a comma. For all Sensor types, enter **8**.

Customize the Analyzer Runtime Environment

The Analyzer can optionally be customized by plugin beans in JAR files.

The location of these JAR files needs to be added to the Analyzer CLASSPATH.

Please specify a semicolon delimited list of any additional JAR files you wish to be added to the CLASSPATH. (for example, 'C:\TVision\myext.jar;C:\TVision\myutil.jar') []:

```
2005-04-14 09:07:02,506 [main] - TransactionVision  
Info(FileUpdated): File "Z:\tvision\media\TVision\config\  
services\Analyzer.properties" has been successfully updated
```

Specify the location of any JAR files that need to be added to the Analyzer CLASSPATH. These JAR files contain plugin beans that customize the Analyzer.

Completion Information

```
2004-08-23 18:14:53,854 [main] - TransactionVision
Info(TVisionSetupInfoSuccess)
: TVisionSetupInfo has completed successfully.
```

All program output and user input are logged to "C:/Program Files/Bristol/TVision/logs/setup.log".

Please run `TVisionSetup` to complete `TransactionVision` setup.

Note:

- `TVisionSetup` needs to be run in a user environment which has the correct database environment set.
- For IBM DB2 this includes checking the `DB2INSTANCE` environment variable setting and that the user has database access rights.
- An IBM DB2 database, required to store `TransactionVision` data, needs to be created before running `TVisionSetup`.
- Refer to your IBM DB2 manual for further information on database setup.

Using `TVisionSetup`

Run `tvisionSetup` once after installation. When only the Analyzer is installed, you can use this utility to do any or all of the following:

- Initialize the database for `TransactionVision`, or verify that the database is initialized
- Verify that WebSphere MQ or TIBCO EMS is set up for `TransactionVision`
- Initialize the `TransactionVision` Analyzer, or verify that the Analyzer is initialized

Before You Begin

Before you run `tvisionSetup`, you must set the correct database environment in your user environment:

- For IBM DB2, the `DB2INSTANCE` environment variable must be set and you must have database access rights. For Oracle, this includes checking that the user has database access rights and, if necessary, that the `ORACLE_HOME` environment variable is set.
- A database to store `TransactionVision` data must be created before running `tvisionSetup`.

Refer to your IBM DB2 or Oracle documentation for further information on database setup.

Required Information

For most tasks, **TVisionSetup** uses information provided in the **TVisionSetupInfo** utility. However, some tasks may require you to enter additional information. Please review these information requirements before running **TVisionSetup** so you will be prepared to enter appropriate responses.

Verify WebSphere MQ Setup

To verify WebSphere MQ setup for TransactionVision, you must supply the following information:

- The name of the WebSphere MQ queue manager you wish to verify
- Whether to test the server connection and/or the client connection. If you test the client connection, you must enter the following:
 - Queue manager host name
 - Queue manager channel name
 - Queue manager port number

Verify TIBCO Setup

To verify TIBCO setup for TransactionVision, you must supply the following information:

- The name of the TIBCO EMS server host you wish to verify
- The port number of the TIBCO EMS server you wish to verify
- The user name you wish to use to connect to the TIBCO EMS server
- The password for the specified user name

Initialize Analyzer

Once the Analyzer is initialized, you must specify whether to start the Analyzer.

TVisionSetup Syntax

Use the following commands to run **TVisionSetup**:

Operating System	Script Command
AIX, Linux, Solaris	<TVISION_HOME>/bin/TVisionSetup.sh

```
Windows <TVISION_HOME>\bin\TVisionSetup.bat
```

The syntax for **TVisionSetup** is as follows:

```
<TVISION_HOME>/bin/TVisionSetup [-rmiregp PORTNUMBER]  
[-debug] [-h]
```

- The default port number where the Analyzer listens for RMI calls is 1099; use the `-rmiregp` option to specify a different port number.
- Use the `-debug` option to start `rmid` and `rmiregistry` in debug mode.
- Use the `-h` option to print the usage message for this utility.

TVisionSetup must be run in a user environment which has the correct database environment set. For IBM DB2 this includes checking the `DB2INSTANCE` environment variable setting and that the user has database access rights. Refer to your IBM DB2 manual for further information.

TVisionSetup Menus

This command provides different menus, depending on whether the host is a WebSphere or WebLogic AppServer with the TransactionVision web user interface installed, and whether LDAP demo mode is used.

Enter the number corresponding to the desired menu action, or enter **99** to perform all actions. The results of each menu action are displayed as they are performed; press the Enter key to continue with the next menu item. Some menu actions require user input.

The following menu appears if WebSphere Application Server and the TransactionVision web user interface are installed, and the DEMO mode is **not** used:

1. Initialize database for TransactionVision
2. Verify database initialization for TransactionVision
3. Verify WebSphere MQ setup for TransactionVision
4. Verify TIBCO setup for TransactionVision
5. Initialize TransactionVision Analyzer
6. Verify TransactionVision Analyzer initialization
99. Select all
- V. Display version info for software

Type "quit" or "q" to exit

TVisionSetup Prompts

The following sections describe the required input for each menu item.

1. Initialize database for TransactionVision

No input required. Note that this menu item always creates the TVision schema in the default tablespace. To use a different tablespace, run the CreateSqlScript script the -ts option.

2. Verify Database initialization for TransactionVision

No input required.

3. Verify WebSphere MQ setup for TransactionVision

This menu item displays the following output and prompts:

```
Verifying WebSphere MQ Java setup for TransactionVision ...
```

```
Please specify WebSphere MQ queue manager name:
```

```
Do you want to test WebSphere MQ server connection? (Y/N)  
[N]:
```

```
TransactionVision Info(MQInitialized): WebSphere MQ is  
loaded properly for WebSphere MQ server connection
```

```
Do you want to test WebSphere MQ client connection? (Y/N)  
[N]:
```

Important! If you answer **Y**, the following you are prompted to enter the following additional information:

```
Host name for WebSphere MQ queue manager:
```

```
Channel name for WebSphere MQ queue manager:
```

```
Port number for WebSphere MQ queue manager:
```

```
Hit <Enter> to continue ...:
```

4. Verify TIBCO sertup for TransactionVision

This menu item displays the following output and prompts:

```
Verifying TIBCO EMS setup for TransactionVision ...
```

```
TIBCO EMS Server Host Name []:
```

```
TIBCO EMS Server Port Number []:
```

```
User Name []:
```

```
Password []:
```

```
Testing TIBCO EMS server connection ...
```

```
2005-04-12 09:21:35,631 [main] - TransactionVision  
Info(TIBCOInitialized): TIBCO is loaded properly for TIBCO
```

```
connection
```

```
Hit <Enter> to continue....:
```

5. Initialize TransactionVision Analyzer

This menu item displays the following output and prompts:

```
Initializing TransactionVision Analyzer ...
```

```
Do you want to start TransactionVision Analyzer? (Y/N) [Y]:
```

```
Hit <Enter> to continue ....:
```

6. Verify TransactionVision Analyzer initialization

No input required.

Additional Analyzer Configuration

The following parameters should be checked for the Analyzer:

- Make sure the DBMS schema has been tuned up and tested. For more information, see Chapter 4, “Configuring Databases.”
- Configuration Message Expiry 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. For more information, see the *TransactionVision Administrator's Guide*.
- Analyzer thread count should be set to match the test results from the DBMS insert test. For more information, see the *TransactionVision Administrator's Guide*.

Windows Service Properties

On Windows, the Analyzer is a Windows service named **TransactionVision Analyzer Service**, which can be managed through the Services administrative tool.

Service Properties

By default, the Analyzer service is configured to use the local system account and to startup automatically when the system starts. To change the account name or other service properties, use the Windows Services administrative tool, accessible through the Control Panel.

Important! Note that the Analyzer must have access to WebSphere MQ and a database. For it to function correctly, the assigned Log On account must have permission to access these resources.

Error Logging

Once the Analyzer is running, it logs its errors to `TVISION_HOME/logs/analyzer.log`. If an error occurs while initializing the service portion of the Analyzer, however, they can be found in the `analyzer_startup.log` file.

Stopping the Analyzer Service Without Stopping Data Collection

There may be times when you wish to stop the Analyzer service, but you not want to stop data collection. To do this, use the following command to stop the Analyzer service:

```
ServicesManager -cmd killserver
```

For more information about **ServicesManager**, see Appendix A.

Multithreaded Servlet/JMS Events

By default, if a servlet spins off a thread to make some JMS calls, the servlet passes tracking information to the child thread. The result is that both the servlet and JMS events belong to the same business transaction. However, there may be some cases in which you wish to separate these events into different transactions. For example, a servlet may spin off a long-running thread that you do not want to be part of the same transaction as the servlet.

To change the default behavior so that the child thread is not considered by `TransactionVision` to be part of the same business transaction as the servlet that spins it, perform the following steps:

1. Open the `<TVISION_HOME>/config/services/Analyzer.properties` file.
2. Add the name of the program that spins off threads to the `separate_child_thread_txns` property, as in the following example:

```
separate_child_thread_txns=program1, program2
```

Separate multiple program names with a comma.
3. Close and save `Analyzer.properties`.
4. Restart the Analyzer.

Local Transaction Matching

By default, the Analyzer uses strict local transaction matching to group WebSphere MQ events into local transactions. However, there may be times when you want to disable strict local transaction matching and instead use the default MQ local transaction bean, which groups events into local transactions according to unit of work.

To disable strict local transaction matching for certain WebSphere MQ events, you must define criteria in the TransactionVision `MQStrictLocalTxnExclude.xml` file. If an event matches any of the criteria specified in this file, the Analyzer will **not** put it into a separate local transaction, but will use the default MQ local transaction bean instead.

Criteria consist of a program name, queue manager, and object name combination. The following example disables strict local transaction matching for events that meet either of the following criteria:

- Program name `amqcrsta`, queue manager `QM1`, and object `TEST.Q1`
- Program name `amqcrsta`, queue manager `ALT_QM`, and object `TEST.Q`

```
<Criteria>
  <Match id="0">
    <value
      xpath="/Event/StdHeader/ProgramName">amqcrsta</value>
    <value
      xpath="/Event/Technology/MQSeries/MQObject/@queueManager">
QM1</value>
    <value
      xpath="/Event/Technology/MQSeries/MQObject/@objectName">
TEST.Q1</value>
    </Match>

    <Match id="1">
      <value
        xpath="/Event/StdHeader/ProgramName">amqcrsta</value>
      <value
        xpath="/Event/Technology/MQSeries/MQObject/@queueManager">
ALT_QM</value>
      <value
        xpath="/Event/Technology/MQSeries/MQObject/@objectName">
TEST.Q1</value>
      </Match>
    </Criteria>
```

Analyzer Failure Mode

The Analyzer can operate in two modes: standard mode and failure mode. In failure mode, the Analyzer will only store event data of failed business transactions (or transactions violating SLA). For successful business transactions, only the corresponding business transaction rows will be saved. Using failure mode reduces the data storage requirement for a project. For more information about enabling failure mode, see the *TransactionVision Administrator's Guide*.

Reducing Event Database Size

TransactionVision provides an XML event compression bean. Use this bean to reduce the database size for each event.

Important! If the XML Event Compression bean is enabled, it is not possible to query on user buffer data.

After running **TVisionSetupInfo**, open the file

<TVISION_HOME>/config/services/Beans.xml, change the following segment:

Original Segment:

```
<Module name="DBWriteEventCtx" type="Context">

  <!-- This context contains beans that write the XML event
(or part of it) to the database. -->
  <!-- Each registered bean in the chain is called. -->

  <Module
class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventDefaultBean" type="Bean"/>

  <!-- Replace the default bean with this one if you want
ZIP compression for the XML event -->
  <!--Module type="Bean"
class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventCompressedBean" /-->
</Module>
```

Modified Segment:

```
<Module name="DBWriteEventCtx" type="Context">

  <!-- This context contains beans that write the XML event
(or part of it) to the database. -->
  <!-- Each registered bean in the chain is called. -->

  <!--<Module
class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventDefaultBean" type="Bean"/-->

  <!-- Replace the default bean with this one if you want
ZIP compression for the XML event -->
  Module type="Bean"
class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventCompressedBean" />
</Module>
```

Important! Make sure only one bean is enabled at one time; otherwise, when both `DBWriteEventDefaultBean` and `DBWriteEventCompressedBean` are enabled, an exception will be thrown.

When using the compression bean, your `DatabaseDef.xml` must be updated to store the event data in BLOB format. The `EVENT` and `EVENT_OVERFLOW` table definitions must be updated, and the `event_data` column type changed from `CLOB` to `BLOB`. After this change has been made you will need to restart your Analyzer and Web application, and create a new project for the changes to take effect. The following example shows `DatabaseDef.xml` before the change to store event data in BLOB format:

```
<Table name="EVENT" volatile="true">
  [...]
  <Column name="event_data" type="CLOB" size="1M"/>
  [...]
</Table>
<Table name="EVENT_OVERFLOW" volatile="true">
  [...]
  <Column name="event_data" type="CLOB" size="1M"
notNull="true"/>
  [...]
</Table>
```

Chapter 6

Configuring Analyzer Logging

Log Files

By default, all TransactionVision components log error and warning messages to the appropriate log files. The location of log files is specified when you run `TVisionSetupInfo` or `SensorSetup` and stored in the `Setup.properties` file.

The Analyzer logs error messages to the `analyzer.log` file. On Windows, the Analyzer uses three additional log files:

- `analyzer_startup.log` contains information about the running of the Windows service portion of the Analyzer. It typically contains information about what options the Analyzer started under. If errors are encountered during the initializing of the service portion of the Analyzer, they can be found in this file.
- `analyzer_stderr.log` and `analyzer_stdout.log` represent the standard output and error of the Analyzer process. If you have custom analysis beans that print to the console or to standard error, you can find their output in these files. These files should also be referred to for further information if you see problems in starting or running the Analyzer and the standard `analyzer.log` file does not contain anything indicating an error.

Circular Logging

By default, the Analyzer employs a form of circular logging. When the log file reaches the configured maximum size, it is renamed as a backup file and a new, empty log file is created. By default, the maximum log size is 10 MB and there is one backup log file.

Using the defaults, when a log file (for example, the Analyzer log file `analyzer.log`, reaches 10 MB in size, it is renamed `analyzer.log.1` and a new `analyzer.log` file is created. If you change the configuration so that there are two backup files, the following events take place when `analyzer.log` reaches 10 MB:

1. `analyzer.log.2` is removed if it exists.
2. `analyzer.log.1` is renamed `analyzer.log.2`.
3. `analyzer.log` is renamed `analyzer.log.1`.
4. A new `analyzer.log` is created.

If you do **not** wish to use circular logging, you may change the configuration to use linear logging, in which a single log file is generated.

The `<TVISION_HOME>/config/logging/*.Logging.xml` files specify the type of logging used, the maximum log file size, and the number of backup log files for each component. For example, `Sensor.Logging.xml` specifies the configuration for the servlet and JMS Sensors. This file contains entries similar to the following:

```
<appender
class="tvision.org.apache.log4j.RollingFileAppender"
name="SENSOR_LOGFILE">
  <param name="File" value="c:/Program
Files/Bristol/TransactionVision/logs/sensor.log"/>
  <param name="Append" value="true"/>
  <param name="MaxBackupIndex" value="2"/>
  <param name="MaxFileSize" value="10MB"/>
  <layout class="tvision.org.apache.log4j. PatternLayout">
    <param name="ConversionPattern" value="%d [%t] %-5p %c
%x - %m%n"/>
  </layout>
</appender>
```

Maximum Log File Size

To change the maximum size of the log file, change the value of the `MaxFileSize` parameter to the desired size. Values provided should end in "MB" or "KB" to distinguish between megabytes and kilobytes.

Maximum Number of Backup Log Files

To change the number of backup files, change the value of the `MaxBackupIndex` parameter to the desired number of backup files.

Changing from Circular to Linear Logging

To use linear logging rather than circular logging, do the following:

1. In the appender class value, change `RollingFileAppender` to `FileAppender`. For example, in the previous example, change the

first line to the following:

```
<appender
class="tvision.org.apache.log4j.FileAppender"
name="SENSOR_LOGFILE">
```

2. Remove the entries for the `MaxBackupIndex` and `MaxFileSize` parameters.

Trace Logging

Trace logging provides verbose information of what a TransactionVision Analyzer is doing internally. It is used mainly to troubleshoot problems and should **not** be turned on in production environments.

To enable trace logging for the TransactionVision Analyzer, set the value of the `trace` property in the

`<TVISION_HOME>/config/services/Analyzer.properties` file to `on`. After modifying this configuration file, you must restart the Analyzer for the change to take effect.

Using Windows and UNIX System Logs

On UNIX and Windows platforms, you can configure TransactionVision to log output to the system event logging facilities—the event log for Windows or `syslog` for UNIX. Examples of the logging configuration files needed to do this can be found in

`TVISION_HOME/config/logging/system/*/Sensor.Logging.xml`.

For both Windows and UNIX, you must define a specialized event appender.

Windows Event Appender

The following example shows how to configure the Windows event appender to use the event log:

```
<appender name="NT_EVENT_LOG"
class="tvision.org.apache.log4j.nt .NTEventLogAppender">
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d [%t] -
%m%n"/>
  </layout>
</appender>
```

`NT_EVENT_LOG` can then be referenced in a category definition of your choice. For example:

```
<category additivity="false"
class="com.bristol.tvision.util. log.XCategory"
name="sensorLog">
```

```
<priority class="com.bristol.tvision.util.log.XPriority"
value="info"/>
<appender-ref ref="NT_EVENT_LOG"/>
</category>
```

On Windows, you must also add a special DLL to your path. This DLL, **NTEventLogAppender.dll**, can be found in the `config\logging\system\bin` directory. For example:

```
set path=%TVISION_HOME%\config\logging\system\bin;%PATH%
```

UNIX Event Appender

The following example shows a UNIX event appender to use syslog:

```
<appender name="SYSLOG"
class="tvision.org.apache.log4j.net.SyslogAppender">
  <param name="SyslogHost" value="localhost"/>
  <param name="Facility" value="local0"/>
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="[%t] %-5p %c
%x
- %m%n"/>
  </layout>
</appender>
```

Specify the `SyslogHost` and `Facility` parameters as appropriate for your environment.

Enabling SMTP Logging

The `SMTPAppender` sends an email to the SMTP server when an error log event at the specified threshold reaches the appender. To enable the `SMTPAppender`, add the following to your `Analyzer.logging.xml` file:

```
<appender name="EMAIL"
class="tvision.org.apache.log4j.net.SMTPAuthenticateAppender">
  <param name="SMTPHost" value="smtp.myserver.net"/>
  <param name="To" value="analyzer_log4j@myserver.net"/>
  <param name="From" value="administrator@myserver.net"/>
  <param name="UserName" value="smtp_user"/>
  <param name="Password" value=""/>
  <param name="Authenticate" value="true"/>
  <param name="BufferSize" value="1"/>
  <param name="Threshold" value="info"/>

  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d [%t] %-
5p %c %x - %m%n"/>
  </layout>
</appender>
```

The `threshold` parameter specifies the logging level that is allowed to append into the SMTPAppender.

To define a customized triggering event evaluator, add the `EvaluatorClass` parameter:

```
<param name="EvaluatorClass" class="tvision.org.apache.log4j.spi.TriggeringEventEvaluator"/>
```

This interface provides the following function for determining when an email should be sent:

```
public boolean isTriggeringEvent(LoggingEvent event) {
    long l = 0;
    synchronized(lock) {
        l = (msgCount ++);
    }
    return (((l + 1)%msgPkgSize) == 0); // fire email on
    every msgPkgSize events.
}
```

Enabling SNMP Logging

The `JoeSNMPTrapSender` appender sends email when a specified error level occurs. To enable `JoeSNMPTrapSender`, add the following definition to the `Analyzer.logging.xml` file:

```
<!-- SNMP TRAP appender !-->
<appender name="TRAP_LOG"
class="tvision.org.apache.log4j.ext.SNMPTrapAppender">
    <param name="ImplementationClassName"
value="tvision.org.apache.log4j.ext.JoeSNMPTrapSender"/>
    <param name="ManagementHost" value="127.0.0.1"/>
    <param name="ManagementHostTrapListenPort" value="162"/>
    <param name="EnterpriseOID" value="1.3.6.1.4.1.24.0"/>
    <param name="LocalIPAddress" value="127.0.0.1"/>
    <param name="LocalTrapSendPort" value="161"/>
    <param name="GenericTrapType" value="6"/>
    <param name="SpecificTrapType" value="12345678"/>
    <param name="CommunityString" value="public"/>
    <param name="ForwardStackTraceWithTrap" value="true"/>
    <param name="Threshold" value="DEBUG"/>
    <param name="ApplicationTrapOID"
value="1.3.6.1.4.1.24.12.10.22.64"/>
    <layout class="tvision.org.apache.log4j.PatternLayout">
        <param name="ConversionPattern"
value="%d,%p,[%t],[%c],%m%n"/>
    </layout>
</appender>
```

Important! You must add `$TVISION_HOME/java/lib/joesnmp.jar` to your `CLASSPATH` because it is required by `JoeSNMPTrapSender`.

Chapter 7

Integrating TransactionVision with Tivoli Enterprise Console

This section describes how TransactionVision integrates with the Tivoli Enterprise Console (TEC). This integration allows TransactionVision to send events into TEC.

Events generated by TransactionVision can be of two types:

- Internal events, generated by TransactionVision components, Analyzer and UI
- Application events, generated by plugin beans in the Analyzer or scheduled jobs in the user interface components of TransactionVision

TEC Integration Setup

Setting up TransactionVision's integration with TEC involves TEC server side setup as well as TransactionVision Analyzer and user interface setup.

TEC Server Side Setup

TEC server side setup enables the reception of TransactionVision events (BTV) into TEC by readying the current TEC rulebase, or by creating a new TEC rulebase by cloning the currently loaded rulebase with the addition of the appropriate BTV class definitions and ruleset. In either scenario, you may load the newly configured rulebase when you run `install.sh`, or at a later time.

TransactionVision Analyzer and User Interface Setup

Error messages as well as audit logs from the TransactionVision Analyzer and user interface components can be routed to TEC. TransactionVision uses the log4j package (<http://logging.apache.org/log4j/docs>) for all logging. By default, these log messages are sent into text files.

TransactionVision provides an appender (TECAppender) to route these messages into TEC.

Installation and Setup

TEC Server Side Setup

Important! Perform the TEC server side setup **before** enabling the TransactionVision Analyzer or user interface for generating TEC events. Otherwise, events of undefined classes can be passed to TEC. The TEC server side setup can be done independently of the TransactionVision side setup.

Installation Requirements:

- Run this Pac at the TEC server.
- Unix UID with privileges to execute into, and write to the TEC rulebase directory tree
- Tivoli Admin with at least "Senior" authorization

Instructions

1. Untar the file `<TVISION_HOME>/tivoli/TVisionTEC.tar` in a scratch directory. The scratch directory will be populated with the following files:
 - `Install.sh`
 - `README`
 - `btv.baroc`
 - `btv.rls`
 - `btv_escalate.sh`
 - `btv_take_action.sh`
2. Run `Install.sh` locally from within that directory. Follow online prompts.
3. If utilizing ITM, download `TVision_ITM_RM.tar` from the directory `<TVISION_HOME>/tivoli` on the Analyzer installation host to the ITM host. Copy it to the resource model tar directory (ex. `/usr/local/Tivoli/bin/lcf_bundle.40/Tmw2k/Rm`) and import into ITM.

After install, the scratch directory and all its contents may be deleted.

Additional Information:

The `btv.rls` ruleset will be loaded by default as "last in line" in the `rule_sets` and `rule_sets_EventServer` datasets. A class definition file (`btv.baroc`) and ruleset (`btv.rls`) will be loaded. If manual loading of these files is preferred, do not run the `Install.sh` script.

The rules are delivered in a "commented out" state, and need to be uncommented and recompiled and loaded for use as desired. Ruleset include dupe detect, 1st action, and escalation rules. The action and escalation scripts should be inserted into the TEC scripts directory. Rules and action scripts need to be tailored to your environment before uncommenting. This does not stop the installation of the class definitions and the readying of the TEC server to accept TransactionVision events.

TransactionVision Analyzer and User Interface Setup

To enable TransactionVision to write events into TEC, you must modify `log4j` configuration files to use the `TECAppender`.

1. Edit the Analyzer `log4j` configuration file

```
<TVISION_HOME>/config/logging/Analyzer.Logging.xml.
```

To enable error logs to be sent into TEC, add the line in bold below:

```
<category name="AppLog"
class="com.bristol.tvision.util.log.XCategory"
additivity="false">
  <priority value="info"
class="com.bristol.tvision.util.log.XPriority"/>
  <appender-ref ref="ANALYZER_LOGFILE" />
  <appender-ref ref="TECAPPENDER" />
</category>
```

To enable audit logs, where every start and stop action is recorded to TEC, add the line in bold below:

```
<category name="AnalyzerActivityLog"
class="com.bristol.tvision.util.log.XCategory"
additivity="false">
  <priority value="info"
class="com.bristol.tvision.util.log.XPriority"/>
  <appender-ref ref="ANALYZER_ACTIVITY_LOGFILE" />
  <appender-ref ref="TECAPPENDER" />
</category>
```

2. Run `<TVISION_HOME>/bin/TVisionSetupInfo.[sh|bat]` again to enter the JAR files `evd.jar` and `tivlog.jar` into the Analyzer `CLASSPATH` setting.
3. Restart the Analyzer for this change to take effect.

4. Edit the UI log4j configuration file

<TVISION_HOME>/config/logging/UI.Logging.xml on the machine you are running the user interface component on your application server.

To enable error logs to be sent into TEC, add the line in bold below:

```
<category additivity="false"
class="com.bristol.tvision.util.log.XCategory"
name="AppLog">
  <priority
class="com.bristol.tvision.util.log.XPriority"
value="info"/>
  <appender-ref ref="UI_LOGFILE"/>
  <b>appender-ref ref="TECAPPENDER"/>
</category>
```

To enable audit logs, where every page action by the user is recorded into TEC, add the line in bold below:

```
<category additivity="false"
class="com.bristol.tvision.util.log.XCategory"
name="UIActivityLog">
  <priority
class="com.bristol.tvision.util.log.XPriority"
value="info"/>
  <appender-ref ref="UI_ACTIVITY_LOGFILE"/>
  <b>appender-ref ref="TECAPPENDER"/>
</category>
```

5. Restart the TransactionVision application loaded within your application server.

6. Edit the file <TVISION_HOME>/config/logging/tivoli/btv_tec.cfg file to point to the TEC server host.

```
BufEvtPath=<TEMPDIR>/btv_tec.cache
LogFileName=<TEMPDIR>/btv_tec.log
LogLevel=ALL
ServerLocation=<TEC server name>
ServerPort=0
```

Errors from the log4j TECAppender go into <TVision LOGDIR>/tecappender.log (typically <TVISION_HOME>/logs/tecappender.log, look in the file <TVISION_HOME>/config/setup/Setup.properties to see where this has been set to).

7. A monitoring profile that allows Tivoli ITM to monitor the healthy functioning of the TransactionVision Analyzer process is also available.

Import the supplied Resource Model (TVisionITM_RM.tar) into ITM (for example, wdmrm - add TVisionITM_RM.tar) and

distribute it as usual with respect to ITM. This Resource Model checks the state of the Analyzer and sends an alert to TEC if the Analyzer is deemed down. It is based on ITM V5.1.1

Modifying the Beans.xml File in Your TransactionVision Installation Directory

You may optionally customize the Analyzer with plugin beans in JAR files. You must add the location of these JAR files to the Analyzer CLASSPATH. Specify a semicolon-delimited list of an additional JAR files you wish to add to the CLASSPATH. For example:

```
'C:\TVision\myext.jar;C:\TVision\myutil.jar' []: C:\Program Files\Bristol\TransactionVision\java\lib\evd.jar;C:\Program Files\Bristol\TransactionVision\java\lib\tivlog.jar
```

where `C:\Program Files\Bristol\TransactionVision` should be replaced by the location of your `TVISION_HOME` and the separator is semicolon on Windows and colon on UNIX.

Appendix A Utilities Reference

CreateSqlScript

Location:

TVISION_HOME/bin/CreateSqlScript.[sh|bat]

Purpose:

Allow the user to create and optionally execute a SQL script to create, drop, import or export TransactionVision system tables or project tables.

Important! The TransactionVision Analyzer and web application need to be stopped before performing any database imports or exports. This is to avoid causing the database import/exports to fail because of database locks held by the Analyzer or web application. If a schema is dropped, make sure that there are no active projects or users logged in that are using that schema.

Syntax:

```
CreateSqlScript
  {-create(-c) | -drop(-d) | -import(-i) | -export(-ex) | -
resetseq(-r) }
  {-system(-s) | -project(-p) SCHEMA | -table(-t) TABLE
SCHEMA}
  [[-noscript(-n)] -execute(-e)]
  [-tablespace(-ts) TABLESPACE]
  [-fileType(-f) IXF|DEL] [-lobPath(-lp) PATH]
  [-dbproperties(-db) FILE]
```

Options:

Option	Description
-drop (-d)	Drop tables
-create (-c)	Create tables

Option	Description
-execute (-e)	Execute script
-noscript (-n)	No script generation
-system (-s)	Create/drop system tables
-project (-p) SCHEMA	Create/drop project tables in schema SCHEMA
-table (-t) TABLE SCHEMA	Create/drop table TABLE in schema SCHEMA.
-tablespace (-ts) TBSPC	Use tablespace TBSPC.
-dbproperties (-db) FILE	Use Database.properties file FILE.
-import (-i)	Generates a database import script. To run database scripts, use the command <code>db2 -n -t -f <sql script filename></code> . For a DB2 database, you can combine this option with the <code>-fileType</code> and <code>-lobPath</code> options to customize the data format and the location of LOB. You may NOT combine this option with the <code>-noscript</code> or <code>-execute</code> options.
-export (-ex)	Generates a database export script. To run database scripts, use the command <code>db2 -n -t -f <sql script filename></code> . For a DB2 database, you can combine this option with the <code>-fileType</code> and <code>-lobPath</code> options to customize the data format and the location of LOB. You may NOT combine this option with the <code>-noscript</code> or <code>-execute</code> options.
-resetseq (-r)	Resets the sequence start number to match to match the imported data.
-fileType (-f) IXF DEL	Specify the DB2 data output file format used for importing/exporting data. The default type is IXF. For the IXF file type, the import/export script uses the LOBFILE option for rows that contain greater than 32K data. For the DEL type, the import/export script exports LOBFILES into a single file (requires FixPack 8).
-lobPath (-lp) PATH	Specifies the directory for DB2 LOBFILES. The default value is the current directory.
-noLob (-nl)	Do not generate DB2 export/import SQL with LOBINFILE option. This option will truncate LOB data to the first 32K bytes.

Examples:

1. Create system tables with schema as TVISION and execute the procedure without generating SQL script:

```
CreateSqlScript -e -n -c -s
```

2. Generate SQL script for creating project tables with schema as PROJECT without executing the procedure:

```
CreateSqlScript -c -p PROJECT
```

3. Drop table EVENT in schema PROJECT and execute the procedure without generating SQL script:

```
CreateSqlScript -e -n -d -t EVENT PROJECT
```

DB2RunStats

Location:

TVISION_HOME/bin/DB2RunStats.[sh|bat]

Description:

Updates statistics about the physical characteristics of a table and the associated indexes. These characteristics include number of records, number of pages, and average record length. The optimizer uses these statistics when determining access paths to the data.

This command is called when a table has had many updates, such as when data is continuously collected into DB2 by the TransactionVision Analyzer. It could result in large performance gains in queries made by TransactionVision views and reports, as well as queries made internally by the TransactionVision Analyzer to correlate events.

This script can be set up to run as a scheduled batch job using either the UNIX `crontab` facility or the Windows scheduler.

Important! While this script is running, TransactionVision Analyzer processing slows down.

This script typically should be run daily, though the frequency of execution could be higher for higher message rates.

Important! This script needs to be customized based on your system to set the correct DB2 installation location and the correct TVISION_HOME location.

Important! If user has additional tables defined for the project schema, new RUNSTATS statements must be added to cover the additional tables.

Syntax:

```
DB2RunStats user_name passwd database_name schema_name [-v7]
```

Options:

Option	Description
user_name	The user account that has privilege to execute RUNSTATS and make database connections
passwd	The password associated with <i>user_name</i>
database_name	The name of the database to connect to

Option	Description
schema_name	The name of the schema that the project reads and writes data to
-v7	Use in a DB2 7.x environment. The default operation is for DB2 8.1.

DB2Test

Location:

com.bristol.tvision.admin.DB2Test

Description:

Measures DB2 database INSERT performance.

The utility inserts sample event data into the RAW_EVENT table of the specified schema. Before running the test, create a new project schema with CreateSqlScript (which can be deleted after running the test).

Syntax:

```
DB2Test databaseName user passwd schema eventCount  
eventSize threadCount {-commit n}{-jdbcBatch}
```

Options:

Option	Description
databaseName	Name of the DB2 Database that contains the schema to be used for the test.
user	DB2 user name
passwd	DB2 password
schema	TransactionVision schema in which sample event data will be saved.
eventCount	Number of events to generate.
threadCount	Number of threads to use to generate events.
-commit n	Executes every <i>n</i> insert statements as part of a single batch operation. Default is to commit each insert individually.
-jdbcBatch	Use JDBC standard batching.

MigrateConfig

Location:

`TVISION_HOME/bin/MigrateConfig.[sh|bat]`

Description:

This is an internal script called during TransactionVision installation to migrate configuration files from an older version of TransactionVision to the current version.

Important! Do NOT call this script directly; it may only be run during installation.

MigrateDB

Location:

`TVISION_HOME/bin/MigrateDB.[sh|bat]`

Description:

Migrates project database files from an older version of TransactionVision to the current version. This script may be run by the TransactionVision installation process, or at a later time. It must be run in configured TransactionVision environment; the `Database.properties` must be set correctly for communication with the database.

Syntax:

`MigrateDB`

OracleRunStats

Location:

TVISION_HOME/bin/OracleRunStats.[sh|bat]

Description:

Important! Oracle 10g has a built-in job scheduler that automatically collects statistics. You do not need to run this script for version 10g.

Updates statistics about the physical characteristics of a table and the associated indexes. These characteristics include number of records, number of pages, and average record length. The optimizer uses these statistics when determining access paths to the data.

This command is called when a table has had many updates, such as when data is continuously collected into Oracle by the TransactionVision Analyzer. It could result in large performance gains in queries made by TransactionVision views and reports, as well as queries made internally by the TransactionVision Analyzer to correlate events.

This script can be set up to run as a scheduled batch job using either the UNIX `crontab` facility or the Windows scheduler.

Important! While this script is running, TransactionVision Analyzer processing slows down.

This script typically should be run daily, though the frequency of execution could be higher for higher message rates.

Run this script from a user account that has privileges to perform database operations. This script must be run under an Oracle user account.

Important! This script needs to be customized based on your system to invoke the correct environment initialization script like `.profile` or `.bashrc`, and to set the correct Oracle installation location and the correct TVISION_HOME location. Additionally, the file `OracleRunStats.sql` must be in the current directory when running this script.

Syntax:

```
OracleRunStats user_name passwd database_name schema_name
```

Options:

Option	Description
user_name	The Oracle user account to run the script under.
passwd	The password associated with <i>user name</i>

Option	Description
database_name	The name of the database to connect to
schema_name	The name of the schema that the project reads and writes data to

OracleTest

Location:

com.bristol.tvision.admin.OracleTest

Description:

Measures Oracle database INSERT performance.

The utility inserts sample event data into the RAW_EVENT table of the specified schema. Before running the test, create a new project schema with CreateSqlScript (which can be deleted after running the test).

Syntax:

```
OracleTest databaseName host port user passwd schema
eventCount eventSize threadCount [-VARCHAR | -BLOB |
-LONGRAW] {-commit n} {-jdbcBatch} {-OracleBatch} {-thin}
{-parallel} {-url URL}
```

Options:

Option	Description
databaseName	Name of the Oracle database that contains the schema to be used for the test.
host	Name of host system on which Oracle server exists
user	Oracle user name
passwd	Oracle password
schema	TransactionVision schema in which sample event data will be saved.
eventCount	Number of events to generate.
eventSize	Size of event user data buffer (default is 1024 bytes)
threadCount	Number of threads to use to generate events.
-VARCHAR	Use this option if the RAW_EVENT table has been created with a VARCHAR column definition.
-BLOB	Use this option if the RAW_EVENT table has been created with a BLOB column definition.
-LONGRAW	Use this option if the RAW_EVENT table has been created with a LONGRAW column definition.
-commit n	Executes every <i>n</i> insert statements as part of a single batch operation. Default is to commit each insert individually.
-jdbcBatch	Use JDBC standard batching.
-OracleBatch	Use Oracle update batching.
-thin	User thin client driver. Default is to use oci client driver.
-parallel	Use the Oracle INSERT PARALLEL option instead of the standard INSERT INTO.
-url URL	By default, OracleTest will use an appropriate JDBC URL for thin or oci client drivers. However the default may be overridden by specifying the JDBC URL here.

ServicesManager

Location:

TVISION_HOME/bin/ServicesManager.[sh|bat]

Purpose:

Manage the TransactionVision Analyzer service. The Analyzer uses an embedded RMI registry so that Analyzers may be controlled by the TransactionVision web user interface running on remote hosts.

Syntax:

```
ServicesManager  
  {(-start [-project (-proj) PROJECTNAME]) |  
   (-stop [-quiesce][[-project (-proj) PROJECTNAME]  
  [-keepcollect]) |  
   (-exit [-quiesce][[-keepcollect]) |  
   (-status [-project (-proj) PROJECTNAME]) |  
   (-killserver) |  
   (-reconfig {classification | logging | analyzer}) |  
   (-versioninfo)  
   ([-host HOST][[-rmiregp PORTNUMBER][[-debug]])
```

Options:

Option	Description
-start	Starts the Analyzer process if it is not already running.
-stop	The Analyzer stops collecting event data.
-exit	The Analyzer stops collecting event data, then the process exits.
-quiesce	The default behavior of the Analyzer on a stop or exit is to immediately close down collection. If this flag is set, the Analyzer clears out any pending events in the event queue before stopping or exiting. Note that if there is a large event backlog, the Analyzer may take some time to stop or exit.
-keepcollect	Do not send stop message to Sensor. Used in combination with the -stop or -exit option so that the Sensor keeps collecting.
-status	Reports the current Analyzer status.

Option	Description
<code>-reconfig</code>	<p>Reloads Analyzer configuration settings (classification/logging/all settings) without stopping event collection.</p> <ul style="list-style-type: none"> <code>classification</code> - Re-initializes the classification rule engine and reloads the Analyzer classification files reading in any changes that may have been made to that file. <code>logging</code> - Reloads the log4j XML configuration files in the directory <code><TVISION_HOME>/config/logging</code>. <code>analyzer</code> - Re-initializes the Analyzer by reloading most Analyzer settings in the <code>Analyzer.properties</code> file, all beans described in the <code>Beans.xml</code> file, re-initializing logging and all XML based rule files. <p>Settings in the "Collection Properties" section of the <code>Analyzer.properties</code> file, except the <code>write_to_buffer_table</code> property, are not reloaded by the Analyzer with the <code>-reconfig</code> option.</p>
<code>-versioninfo</code>	Returns Analyzer version information.
<code>-killserver</code>	Shuts down the Analyzer immediately. This flag is not recommended; <code>-exit</code> is the preferred method for shutting down the Analyzer cleanly.
<code>-project (-proj) PROJECTNAME</code>	Name of the project for the specified command. If the project name contains a space, enclose the project name in double quotation marks (for example, <code>-proj "Project Name"</code>). The project must have a communication link in order for the Analyzer to process events. Only the <code>-start</code> , <code>-stop</code> , and <code>-status</code> commands be performed on a single project. This option cannot be used in combination with the <code>-host</code> or <code>-rmiregp</code> options. When the <code>-project</code> option is used, the Analyzer host and port is looked up from the database.
<code>-host HOST</code>	Name of the host where the Analyzer runs. Can be either name or IP address. Local host is used if not specified.

Option	Description
<code>-rmiregp PORTNUMBER</code>	Port number on which the Analyzer listens for RMI connections. The default value is the value specified by the <code>analyzer_port</code> property in the <code>Analyzer.properties</code> file. This option only takes effect when communicating with an Analyzer that is already running; the Analyzer always uses the value specified in <code>Analyzer.properties</code> when starting up.
<code>-debug</code>	Start the Analyzer process in debug mode.

Examples:

1. Start project PROJECT:
`ServicesManager -start -proj PROJECT`
2. Stop Analyzer on host HOST:
`ServicesManager -stop -host HOST`

TVisionSetup

Location:

TVISION_HOME/bin/TVisionSetup.[sh|bat]

Purpose:

This should be run once after installation. This program helps users setup the LDAP, database, TransactionVision services and WebSphere application.

Different options are presented based on which options are installed and/or running. The following menu appears if WebSphere Application Server and the TransactionVision web user interface are installed, and the DEMO mode is **not** used:

1. Initialize database for TransactionVision
2. Verify database initialization for TransactionVision

3. Verify WebSphere MQ setup for TransactionVision
4. Verify TIBCO setup for TransactionVision

5. Initialize TransactionVision Analyzer
6. Verify TransactionVision Analyzer initialization

7. Initialize LDAP entries for TransactionVision
8. Verify LDAP initialization for TransactionVision

9. Install TransactionVision web application
10. Verify TransactionVision web application installation

99. Select all
- V. Display version info for software

Type "quit" or "q" to exit

If demo mode is used, or the WebSphere application server or TransactionVision web user interface are not installed, certain items are omitted from the menu. If an item is omitted, the remaining items are renumbered in sequence.

- If you specified in `TVisionSetupInfo` to use DEMO mode for LDAP, items 7 and 8 are omitted from the menu.
- If the WebSphere application server or TransactionVision web user interface are not installed, items 7, 8, 9, and 10 are omitted from the menu.

Syntax:

```
TVisionSetup [-rmiregp PORTNUMBER] [-installweb] [-cleanweb] [-h]
```

Options:

Option	Description
-rmiregp PORTNUMBER	Port number on which the Analyzer starts. Default to 1099 if not specified.
-installweb	Install the TransactionVision Web Application on the Application Server.
-cleanweb	Uninstall the TransactionVision Web Application from the Application Server.
-h	Displays a usage message.

TVisionSetupInfo

Location:

TVISION_HOME/bin/TVisionSetupInfo.[sh|bat]

Purpose:

Collects information from the administrator about external tools used by TransactionVision so that CLASSPATH and library path can be setup correctly before running TransactionVision. This utility performs the following:

- Saves installation path for software tools in TVISION_HOME/config/setup/DefaultInstallPath.xml.
- Modifies TVISION_HOME/config/datamgr/Database.properties based on user input.
- Modifies TVISION_HOME/config/ldap/Ldap.properties based on user input.
- If you are using the WebSphere application server, modifies TVISION_HOME/config/websphere/tvision-ws-config.properties based on user input.
- If you are using the WebLogic application server, modifies TVISION_HOME/config/websphere/tvision-wl-config.properties based on user input.
- Generates TVISION_HOME/bin/SetupEnv.[sh|bat] which sets the minimum set of environment variables (JAVA_HOME, CLASSPATH, shared library path for different platforms, etc.) required by TVision.
- If you are using the WebSphere application server, generates TVISION_HOME/install/websphere/WSSetupEnv.[sh|bat] to setup command line parameters for importing different .xml files to XMLConfig.

Syntax:

TVisionSetupInfo.[sh|bat]

Notes:

TVISION_HOME/config/setup/Setup.properties is used to specify the XML file that stores the default software installation path and setup logging file.

User is free to modify any modified/generated files later on.

Since this utility modifies TransactionVision's configuration files, the user needs to have file modification privilege to run it (root on Unix or Administrator on Windows).

The format of `TVISION_HOME/config/setup/DefaultInstallPath.xml` is as follows:

```
<?xml version="1.0"?>
<DefaultInstallPath>
  <OS name="Windows">
    <DB2>C:\Program Files\IBM\sqlllib</DB2>
    <WebSphereMQ>C:\Program Files\IBM\WebSphere
MQ</WebSphereMQ>
    <WebSphereMQJava>C:\Program Files\IBM\WebSphere
MQ\Java</WebSphereMQJava>
    <WebSphereAppServer>C:\Program Files\Websphere\
AppServer</WebSphereAppServer>
    <WebLogic>C:\bea\weblogic81</WebLogic>
  </OS>
  <OS name="SunOS">
    <DB2>/opt/IBMdb2/V7.1</DB2>
    <Oracle></Oracle>
    <WebSphereMQ>/opt/mqm</WebSphereMQ>
    <WebSphereMQJava>/opt/mqm/java</WebSphereMQJava>
    <WebSphereAppServer>/opt/WebSphere/AppServer
</WebSphereAppServer>
<WebLogic>/usr/local/bea/weblogic81</WebLogic>
  </OS>
  <OS name="AIX">
    <DB2>/usr/lpp/db2_07_01</DB2>
    <Oracle></Oracle>
    <WebSphereMQ>/usr/lpp/mqm</WebSphereMQ>
    <WebSphereMQJava>/usr/lpp/mqm/java</WebSphereMQJava>
    <WebSphereAppServer>/usr/WebSphere/AppServer
</WebSphereAppServer>
<WebLogic></WebLogic>
  </OS>
  <OS name="Linux">
    <DB2>/usr/IBMdb2/V7.1</DB2>
    <Oracle></Oracle>
    <WebSphereMQ>/opt/mqm</WebSphereMQ>
    <WebSphereMQJava>/opt/mqm/java</WebSphereMQJava>
    <WebSphereAppServer>/opt/WebSphere/AppServer
</WebSphereAppServer>
    <WebLogic></WebLogic>
  </OS>
</DefaultInstallPath>
```

Appendix B

Configuration Files

The TransactionVision setup utilities save Analyzer configuration information necessary for TransactionVision to operate in the following configuration files. You may also modify these files directly if you need to make any changes to your configuration.

Important! If you modify property files, you must restart the associated application for you changes to take effect. To restart the Analyzer on Windows, use the Windows Services management control panel.

Analyzer.properties

The `<TVISION_HOME>/config/services/Analyzer.properties` file provides general, collection, and analysis configuration information for the TransactionVision Analyzer.

General Properties

- `rmi_client_timeout`
This entry specifies the timeout for the RMI client in minutes. The default value is 2.
- `logging_xml`
Specifies the name of the logging configuration file used by the Analyzer. The default value is `Analyzer.Logging.xml`.
- `trace`
This property specifies whether Analyzer trace logging is on or off. The default value is off. For more information about trace logging, see “Trace Logging 49.”
- `service_jvm_flags`
Specifies any additional JFM flags you want the Analyzer to run with.

- `Service_additional_classpath`
Specifies the classpaths for any custom beans to be used with your Analyzer.
- `jvm_dll`
`service_classpath`
`service_libpath`
On Windows, these properties are automatically generated by **TVisionSetupInfo**. It should not be necessary to change them. If you do change them, note that they will be reset if **TVisionSetupInfo** is run again. These properties have no effect on the TransactionVision environment other than the Analyzer running as a Windows service.
- `analyzer_port`
This entry specifies the default port number that the Analyzer runs on. The default value is 1099. The Analyzer always uses this port number when it starts. The **SerciesManager**. `[bat|sh] -rmiregp` command can be used to set the port number for an Analyzer that is already running.
- `time_server`
Set to **on** to run the time server withing the Analyzer. The time server is required for TIBCO EMS and generic JMS communication link types to calculate the time skew information. To run the time server on a different host, use the **TimeServer**. `[sh|bat]` utility.
- `time_server_port`
Specifies the listening port for the time server to use, if `time_server` is set to on. The default port is 9037.

Collection Properties

The following properties are used for the event collection operations of the Analyzer:

- `batch_commit_count`
Specifies the number of events to batch before a commit is issued. The default value is 50.
- `commit_time_threshold`
Specifies the amount of time, in seconds, after which a commit is forced. The default value is 1.
- `commit_byte_threshold`
Specifies the number of bytes after which a commit is forced. The default value is 1,000,000.
- `write_to_buffer_table`
Specifies whether to write a copy of raw events from the queue

into the RAW_EVENT table in addition to normal processing. The default value is False. This property should never be set to True if `read_from_buffer_table` is also set to True. Use this property to test custom beans. It should always be set to False in a production environment. Note that when you process the events from the RAW_EVENT table, the Analyzer will only process events if the project has one communication link. If you assign more than one communication link to the project, you will get duplicate events.

- `read_from_buffer_table`
Specifies whether to pull any events from the RAW_EVENT table in addition to reading from the queue. The default value is False. This property should never be set to True if `write_to_buffer_table` is also set to True. Use this property to test custom beans. It should always be set to False in a production environment.
- `keep_events`
Specifies whether the Analyzer should delete raw events after they have been successfully processed. Set it to **true** to keep the raw events in the RAW_EVENT table. Their `event_status` column will be set to PROCESSED so that they are not processed again. If you want to process the events again (for example, with a custom Java bean for a different type of analysis), you must first set the `event_status` to NEW. Set it to **false** (the default value) for the Analyzer to delete raw events from the RAW_EVENT table after it processes them.
- `fail_safe_collection_shutdown`
If true, causes the Analyzer to stop sending configuration messages if a serious failure occurs in event collection and processing. The default value is true.
- `jdbc_batching`
The default `jdbc_batching` value (**on**) causes TransactionVision to execute database statements in batch mode.
- `jdbc_batch_count`
Specifies The number of SQL operations to batch in JDBC batching mode. Note that this number should be equal to or a multiple of the `batch_commit_count` property. The default value is 50.

Analysis Properties

The following properties are used for the event analysis operations of the Analyzer:

- `minimal_event_document`
Specifies whether to create a minimal event document. This setting only affects WebSphere MQ event data. The default is off.
- `save_event_document`
Specifies whether to save event documents in the database. The default is on. If this property is set to off, event detail is not available. Furthermore, the Analyzer will not be able to recover asynchronously flushed transaction and static topology data. This means that in the event of a crash or other abnormal program termination, transaction data and statistics data for the static topology view may no longer be accurate.
- `eventmatching_interval`
Specifies the time in seconds to wait between every invocation of event matching. The default is value is 600.
- `partial_event_lifetime`
The maximum time in minutes a partial event entry may exist in the `partial_event` table. When this time limit is reached, the partial event will be flushed. The default value is 10 minutes.
- `latency_resolution`
The resolution used to calculate latency between events. Possible values are:

Value	Description
1	Seconds
10	1/10 seconds
100	1/100 seconds (default)
1000	Milliseconds

- `separate_child_thread_txns`
By default, if a servlet spins off a thread to make some JMS calls, the servlet passes tracking information to the child thread. The result is that both the servlet and JMS events belong to the same business transaction. However, there may be some cases in which you wish to separate these events into different transactions. For example, a servlet may spin off a long-running thread that you do not want to be part of the same transaction as the servlet. If you do **not** want threads spun off by a servlet to be included in the same business transaction as the servlet, list the servlet program name as the value for this property. Separate multiple program names with a comma, as in the following example:

```
separate_child_thread_txns=program1, program2
```

- `enable_dbcaching`
Enables standard mode in addition to or instead of failure mode. If

all of your projects only use Failure Mode, the Standard Mode can be disabled to save some system resources.

- `enable_failure_mode`
Enables failure mode so that the Analyzer will only store event data of *failed* business transactions (or transactions violating SLA). For successful business transactions, only the corresponding business transaction rows will be saved. Since this mode relies heavily on a meaningful value of the 'result' attribute of the business transaction, it requires a transaction classification with suitable rules for the 'result' column to be in place. The failure mode can be enabled on a per-project basis, allowing it to run certain projects in failure mode while running others in the conventional Standard Mode. The default value is off. For more information about failure mode, see the *TransactionVision Administrator's Guide*.
- `dbcache_thread_count`
Specifies the number of threads to use for writing the cached analysis data to the database. The best value is dependent on the hardware the Analyzer is running on and can only be determined by performance measuring with different values, but a good starting point would be to set this number to half the number of collection threads used for a particular schema. The default value is 2.
- `failure_mode_thread_count`
Defines the number of flushing threads to use for Failure Mode. This number should be set high enough so that the cache data can be flushed fast enough, but low enough to not waste too many system resources. The default value is 2.
- `failure_mode_process_delay`
Defines the amount of time, in milli-seconds, which gets added to the SLA value for determining the age out timeout and is a crucial value for the Failure Mode. If this value is set too low, a lot of successful transaction data will unnecessarily age out and be written to the database, which will result in poorer performance, and wasted database disk space. If this value is set too high, too much data will be held in the memory caches. If the cache sizes are not large enough for the system load, cache overflows will occur frequently and lead to very poor performance. The default value is 5 seconds.
- `failure_mode_discard_overflow`
In case of a cache overflow all the data in the caches will normally be written to the database. If the overflow was caused, for whatever reason, by a longer lasting system slowdown, the constant flushing of the cache could further worsen the situation and slow down the Analyzer even more. If the main concern is to

avoid any chance of filling up the event queue in such situations, you can set the above parameter to true and force the discarding of all cache data in case of an overflow. The default value is **off**.

CacheSize.properties

The `<TVISION_HOME>/config/services/CacheSize.properties` file specifies the size of the caches used in the event analysis. A bigger cache size minimizes database access and increases performance, but also increases the amount of memory used by TransactionVision. It sets the following properties:

- `system_model_objects`
Specifies the number of system model object other than PII cached.
- `pii_objects`
Specifies the number of PII system model objects cached.
- `event_based`
Specifies the number of events cached.
- `transaction_based`
Specifies the number of transactions cached.

Database.properties

The `<TVISION_HOME>/config/datamgr/Database.properties` file specifies the database the Analyzer and the TransactionVision web application read from and write to.

Required Entries

The following mandatory entries are required:

- `jdbc_driver=COM.ibm.db2.jdbc.app.DB2Driver`
This entry is the class name of the JDBC driver used. The default for the IBM DB2 JDBC driver is as above. For Oracle, the default is `oracle.jdbc.driver.OracleDriver`.
- `database_connection_name`
For DB2, this entry is the name of the database connection to be used (typically the database alias). For Oracle, this entry is the database name for an oci client connection or the SID (system identifier) for a thin client connection.
- `database_name`
For DB2, this entry is the name of the database on the server to connect to. This name may be different from the

"`database_connection_name`" if a client database connection is used. For Oracle, this entry is the SID.

- `database_host`
This entry is the host the database server is running on.
- `database_port`
Specifies the port number for the Oracle listener on the target host. The default value is 1521. This attribute is only used if you are using the Oracle thin client. For more information about using Oracle with TransactionVision, see “Connecting to an Oracle Database22.”
- `oracle_client_type`
Specifies whether to use the Oracle thin or client JDBC driver. Set it to `thin` (recommended) or `oci` for Oracle 9 or `oci8` for Oracle 8.1.7. The default is `thin`. This attribute is only used if you are using an Oracle database. For more information about using Oracle with TransactionVision, see “Connecting to an Oracle Database22.”
- `db2_instance_env`
This entry specifies the value of the DB2 environment variable DB2INSTANCE.

Optional Entries that Override Automatic Detection

The following optional entries may be setup in this file. These entries override automatic detection.

- `connection_type`
This entry defines how to obtain JDBC connections. Use one of the following values:
 - `JDBC`: Uses connections obtained from the JDBC driver, with no driver connection pooling (default).
 - `DB2DataSource`: Uses DB2 connection pooling.
 - `OracleDataSource`: Uses Oracle connection pooling.
 - `JNDI`: Uses the data source registered in JNDI. To use a JNDI connection, you must also set the `jndi_url` property.
- `database_url`
This entry defines a custom URL to use with the JDBC driver manager.
- `jndi_url`
This entry defines the JNDI name for the database (defined in WebSphere) when using JNDI connections.

- `unicode_db`
If this property is set, all character-based XDM columns with the attribute `unicode=true` will be generated with double the byte size to allow the specified number of characters to be stored in the database.

Optional Entries with Default Values

The following optional entries may be setup in this file. If they are not specified, default values are used.

- `user`
This optional entry is the user name to be used while making the database connection. If this field is empty, the currently logged in user is used to make the database connection. Make sure that the user specified or currently logged has privileges for database access.
- `passwd`
This optional entry is the password to be used while making the database connection. If this field is empty, the currently logged in user's password is used to make the database connection.
- `jce_provider`
This optional entry specifies the Java Cryptographic Extension (JCE) package name to encrypt the database password. The following table shows the valid package names. If this entry is blank, TransactionVision stores the password as plain text.

Provider	Package Name
Sun	<code>com.sun.crypto.provider.SunJCE</code>
IBM	<code>com.ibm.crypto.provider.IBMJCE</code>

The `TVision_SetupInfo` utility automatically searches for an installed JCE provider that supports the DES encryption algorithm. If such a provider is found, `TVision_SetupInfo` sets the `jce_provider` value to the class name of the JCE provider. If no JCE provider is found, `TVision_SetupInfo` displays the following message:

```
Java Cryptography Extension (JCE) is not present in
your current JDK. Password encryption feature will be
disabled and stored in plain text.
```

- `reconnect_interval`
This entry specifies the frequency in seconds that TransactionVision should try to reconnect with the database. The default value is 10.

- `reconnect_timeout`
This entry specifies the amount of time in seconds that TransactionVision should try to reconnect to the database (at the `reconnect_interval`). The default value is 600.

License.properties

The `<TVISION_HOME>/config/license/License.properties` file specifies the TransactionVision license code supplied by Bristol Technology.

Performance.properties

The `<TVISION_HOME>/config/services/Performance.properties` file contains the following settings for performance logging:

- `performance`
Specifies whether to performance logging is on or off. The default is off.
- `count_interval`
Specifies the number of events after which performance data is logged. The default is 500.
- `detail`
Specifies whether to generate detailed performance and statistics logs. The default is off.
- `start_at`
Defines how many events have to get processed before the performance logging will start. The default is 500.
- `get_queuedepth`
Specifies whether to retrieve the current event queue depth for each log interval. The default is off.

Setup.properties

The `<TVISION_HOME>/config/setup/Setup.properties` file specifies the following properties:

- `default_tool_install_path`
The directory location of the `DefaultInstallPath.xml` file, which lists the locations of software components required by TransactionVision
- `logs_dir`
The name of the directory in which to store log files

- `logging_xml`
The name of the logging configuration file
- `minimum_java_version`
The minimum Java version required by TransactionVision
- `minimum_java_version_sun`
The minimum Java version required by TransactionVision on the Solaris platform
- `maximum_java_version`
The highest Java version supported by TransactionVision

Index

A

analyzer
 configuration, 29, 42
 correlating multithreaded servlet/JMS events,
 43
 error logging, 43
 Windows service properties, 42
analyzer.log, 43, 47
Analyzer.properties, 77
analyzer_startup.log, 43, 47
analyzer_stderr.log, 47
analyzer_stdout.log, 47
APP CTL HEAP SZ, 20
APPLHEAPSZ, 20

B

Beans.xml, 45
Bristol Support, contacting, 4
bufferpool, 22

C

CacheSize.properties, 82
circular logging, 47
configuration files, 77
 backup, 8
 migration, 7
contacting Bristol Support, 4
CreateSqlScript, 59

D

database configuration, 17, 82
database schemas
 migration, 7
Database.properties file, 28, 82
databases
 storing unicode data, 28
DB2 bufferpool, 22
DB2 variable settings, 20
DB2_RR_TO_RS, 21
DB2INSTANCE environment variable, 31, 40
DB2RunStats, 26, 62
DB2Test, 24, 64
DBMS configuration and tuning, 23
DBMS performance, 24
DBWriteEventCompressedBean, 46
DBWriteEventDefaultBean, 46
disabling strict local transaction matching, 43

E

event appender
 UNIX, 50
 Windows, 49
event compression bean, 45
event database
 reducing size, 45

I

installation
 packages, 2
 uninstalling on UNIX, 9
 uninstalling on Windows, 15
 UNIX, 5
 UNIX files, 8
 upgrading, 3
 Windows, 11

J

JMS sensor
 correlating multithreaded events, 43

L

License.properties file, 85
linear logging, 48
local transactions
 disabling strict matching, 43
LOCKLIST, 21
logging, 47
 circular, 47
 linear, 48
 SMTP, 50
 SNMP, 51
 trace, 49

M

MAXAPPL, 20
MigrateConfig, 65
MigrateDB, 8, 66
migrating configuration files and database
 schemas, 7

N

NT_EVENT_LOG, 49

O

Oracle database connection, 22
Oracle variable settings, 22
OracleRunStats, 26, 67
OracleTest, 24, 69

P

Performance.properties, 85

R

RUNSTATS, 26
 DB2, 62
 Oracle, 67

S

ServicesManager, 70
Servlet sensor
 correlating multithreaded events, 43
Setup.properties file, 85
SMTP logging, 50
SNMP logging, 51
strict local transaction matching, 43
support, contacting, 4
SYSLOG, 50

T

TECAppender, 53, 55
Tivoli Enterprise Console (TEC)
 integrating TransactionVision with, 53
 logging, 53
 logging, 55
 server side setup, 53, 54
trace logging, 49
TVISION_HOME, 30
TVisionSetup, 73
 menus, 40
 overview, 38
 required information, 39
 syntax, 39
TVisionSetupInfo, 75
 overview, 30
 required information, 30
 syntax, 33

U

unicode data, 28
UNIX
 event appender, 50
 installation, 5
 installation files, 8
 uninstalling, 9
upgrading from previous releases, 3

W

Windows
 event appender, 49

 installation, 11
 uninstalling, 15
Windows Service administrative tool, 42