TransactionVision®

Administrator's Guide
Version 4.2.1

Printed March 31, 2005

Part No. TV10050331 Version 4.2.1 SupportPac A

Contents

TransactionVision Administration Overview

TransactionVision is the transaction tracking solution that graphically shows you the interaction between all the components of your system. TransactionVision non-intrusively records individual electronic events generated by a transaction flowing through a computer network. More importantly, TransactionVision's patent-pending "Transaction Constructor" algorithm assembles those events into a single coherent business transaction.

Graphical analysis of business transactions enable you to:

- Find lost transactions

- Monitor and meet service level agreements

- Improve efficiencies of your business processes

TransactionVision Basics

To understand the tasks required to administer TransactionVision, you must understand the TransactionVision components, as well as some basic concepts.

Components

TransactionVision consists of three major components:

- Sensors

- Analyzers

- TransactionVision Web User Interface

The following diagram shows the relationship between these components:

```
┌─────────────────────────────────────────────────────┐
│          TransactionVision Web User Interface         │
└─────────────────────────────────────────────────────┘
        ↑↓                              ↑↓

┌───────────────┐              ╭──────────────────────╮
│               │              │                      │
│    Analyzer   │  ←─────────→ │      Database         │
│               │              │    Project Tables     │
│               │              │                      │
└───────────────┘              ╰──────────────────────╯
        ↑
        └──────────┬──────────────┬──────────────┐
                   ↓              ↓              ↓
             ┌─────────┐   ┌─────────┐        ┌─────────┐
             │ Sensor  │   │ Sensor  │  ...   │ Sensor  │
             └─────────┘   └─────────┘        └─────────┘
```
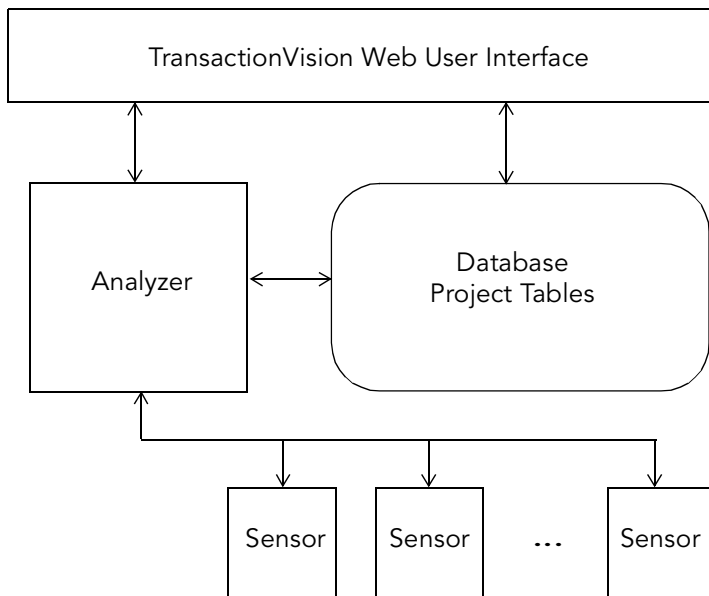
Sensors

Sensors collect transactional events from the various applications involved in your distributed transactions. Sensors are lightweight libraries or exit programs that are installed on each computer in your environment. Each Sensor monitors calls made by supporting technologies on that system and compares them against filter conditions. If the call matches the filter conditions, the Sensor collects entry information about the call, then passes the call on to the appropriate library for processing. When the call returns, the Sensor collects exit information about then call. It then combines the entry and exit information into a TransactionVision event, which it forwards to the Analyzer by placing it on a designated event queue.

TransactionVision provides several types of Sensors:

- WebSphere MQ Sensors
- Servlet Sensor
- JMS Sensor
- EJB Sensor
- CICS Sensor

WebSphere MQ Sensors

The **WebSphere MQ Sensor** tracks MQ API calls. These API calls include the entire MQ API set, the major APIs being MQPUT, MQGET, MQCONN, MQDISC, MQOPEN, MQCLOSE, etc. There are two types of WebSphere MQ Sensors provided by TransactionVision on distributed platforms: the WebSphere MQ Library Sensor and the WebSphere MQ API Exit Sensor. Both of these Sensors report the same information from an MQ API call. They differ primarily in the mechanism by which they intercept MQ API calls, their usage, and the amount of data they collect from the system. For a complete list of monitored APIs, see Appendix D.

- The **WebSphere MQ Library Sensor** intercepts a WebSphere MQ API call by the shared library (or DLL) interception method on distributed platforms. This involves placing the TransactionVision Sensor libraries before the WebSphere MQ libraries in the application library path. This method is useful if you need to track MQ APIs for specific applications.

- The **WebSphere MQ API Exit Sensor** uses the WebSphere MQ API exit support available on distributed platforms in WebSphere MQ v5.3 and later. This Sensor is registered as an exit to the queue manager and invoked when any program connecting to the queue manager invokes a WebSphere MQ API. This method is recommended to collect MQ events from all applications on a queue manager and in particular the listener and the channel agents.

- **z/OS WebSphere MQ Sensors** are provided for tracking MQI API calls in the CICS, batch and IMS environments on the IBM z/OS system. In the CICS environment, the API crossing exit provided by the CICS adapter for WebSphere MQ is used to intercept the MQ API. In the batch and IMS environments, the application has to be re-bound with the Sensor to intercept MQ API calls.

The following supplemental Sensors are available for WebSphere MQ:

- The **Proxy Sensor** correlates business transactions into process that are not monitored using the TransactionVision Sensor libraries (for example, events between a sensored application and an application running on a system where no Sensor is installed such as an external partner system).

- The **WebSphere Business Integration Sensor** (previously known as the MQSI Sensor) distinguishes the various message flows and identifies individual logical transaction paths within WBI. This Sensor is a WBI plugin that provides a trace node, which is inserted into the normal execution path of a message flow, and a failure node, that is inserted into the failure path of a message flow. These nodes generate a MQSI2TRACE event that allows tracking of the message flow within WBI.

- The **WebSphere MQ-IMS Bridge Sensor** tracks WebSphere MQ-IMS bridge messages rather than the WebSphere MQ API calls made by the calling applications. The MQ-IMS Bridge is a component that enables WebSphere MQ applications to invoke IMS transactions and receive their reply messages. The MQ-IMS Bridge Sensor tracks MQ messages coming into the bridge and correlates them with the reply received from IMS. Two events, MQIMS_BRIDGE_ENTRY and MQIMS_BRIDGE_EXIT are generated for every message coming in and going out of the bridge. These events contain the MQ message header and information about which IMS transaction is invoked.

Servlet Sensor

The **Servlet Sensor** tracks servlet methods in a J2EE application server. This Sensor tracks HTTP calls such as HTTP_POST, HTTP_GET, HTTP_PUT, etc., which result in method calls into the J2EE container. The Servlet Sensor tracks these method invocations by instrumenting the servlet to collect events at the entry and exit of each call. For a complete list of monitored APIs, see Appendix D.

JMS Sensor

The **JMS Sensor** tracks WebSphere MQ Java Message Service events from standalone Java applications as well as from J2EE application servers. This Sensor tracks JMS interface methods such as send, receive, etc. These methods are tracked by instrumenting the JMS library to collect events at the entry and exit of each call. For a complete list of monitored methods, see Appendix D.

EJB Sensor

The **EJB Sensor** tracks transactions through business logic within a J2EE application server. This Sensor tracks all public business methods in an entity bean, session bean or message driven bean. In addition to the business methods, this Sensor tracks the ejbCreate, ejbPostCreate, ejbRemove, ejbLoad, ejbStore and onMessage methods. These methods are instrumented by the Sensor to collect events at the entry and exit of each call. For a complete list of monitored methods, see Appendix D.

CICS Sensor

The **CICS Sensor** collects non-WebSphere MQ CICS events to track transactions in a mainframe environment. The CICS Sensor collects data for five types of events: file control, temporary storage, transient data, interval control, and program control. For all types, it tracks information such as Transaction ID, User ID, Terminal ID and SYSID. Other information collected depends on the event type. For a complete list of collected information, see Appendix D.

In the following diagram, shaded areas represent the parts of a web application for which TransactionVision can track events:
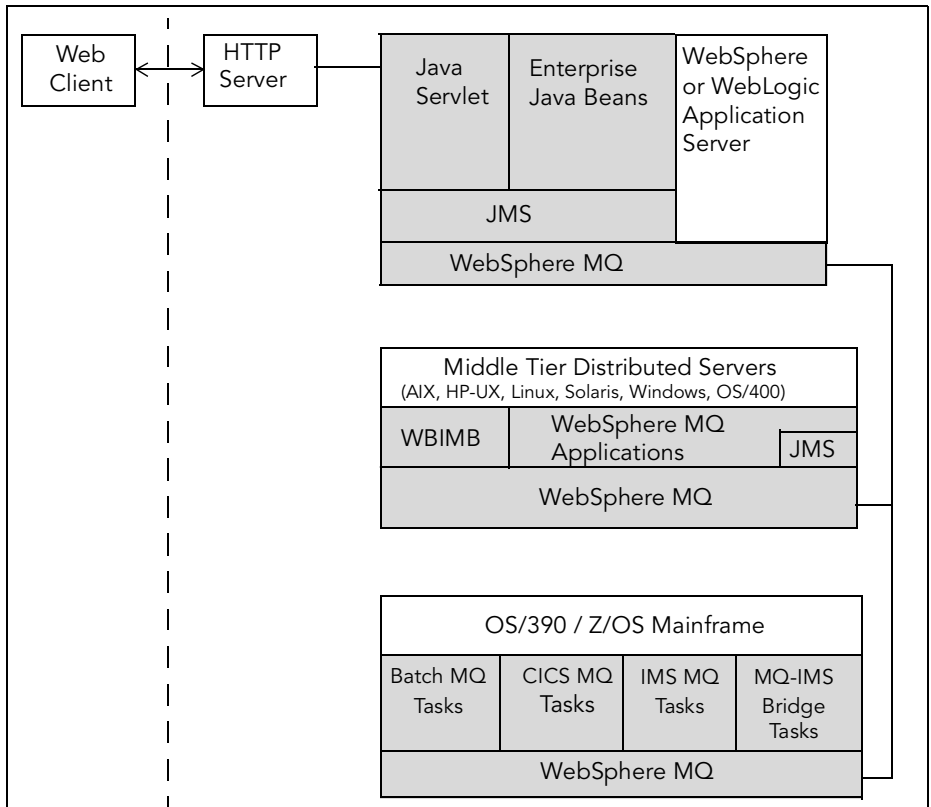
Analyzer

The Analyzer is a service that communicates with Sensors via WebSphere MQ. 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.

Web User Interface

The TransactionVision web user interface is an enterprise application for IBM WebSphere or BEA WebLogic that provides the TransactionVision graphical interface. All interaction is done through web pages. Users and administrators login to the web user interface through a web browser. The web user interface communicates with the Analyzer to provide data collection configuration information such as communication links and data collection filters. It also connects to project database schemas to display project analysis and report results.

TransactionVision Terms and Concepts

To administer TransactionVision effectively, you must be familiar with the following terms and concepts:

- Communication links

- Event Collection

- Data collection filters

- Projects

- Database Schemas

Communication Links

Communication links enable a TransactionVision Sensor to communicate from a host on which an application is being monitored to the Analyzer. Two communication paths must be defined for each communication link: one path for configuration messages from the Analyzer to the Sensors and another path for captured events from the Sensors to the Analyzer. When you define a communication link, you specify the name of the queue manager and queues the Sensor monitors for configuration messages and sends event messages to, as well as the queue managers and queues the Analyzer sends configuration messages to and retrieves event messages from.

Event Collection

The WebSphere MQ Sensor library implements all API entry points and has the same name as the standard library for the monitored technology (for example, `mqm.dll` for WebSphere MQ on Windows). It is installed in a different directory location, and your library search path is altered so that programs load the Sensor library at runtime instead of the standard technology library.

When a program running on the system where a Sensor is installed calls a WebSphere MQ API for the monitored technology, it actually calls the corresponding function in the Sensor library.

The Sensor function first generates a TransactionVision event, recording the API call and other details based on the data collection filters. It then invokes the actual API from the standard technology library. When the actual API returns, the Sensor function generates another TransactionVision event, this time representing the exit state of the function, and then forwards the return information to the calling program.

Important! For WebSphere MQ Sensors to intercept API calls from applications, the applications must link dynamically to the technology library as a shared library. Otherwise, Sensors cannot record events for the applications.

On the OS/390 CICS platform, WebSphere MQ Sensors use the API-crossing exit mechanism provided by the CICS adapter of WebSphere MQ for OS/390. On OS/390 batch and IMS platforms, applications are statically linked to, or dynamically invoke, TransactionVision stubs, which are replacements for the standard WebSphere MQ stubs.

A WebSphere API Exit Sensor is also available for other platforms.

The WebSphere Application Server servlet, JMS, and EJB Sensors use Java bytecode instrumentation to intercept servlet, JMS, and EJB API calls. At installation of the WebSphere servlet Sensor, the server JVM settings are modified to add a Sensor classloader plugin. This plugin intercepts servlet API calls and allows the TransactionVision Sensor to report them. The JMS API called are trapped using static bytecode instrumentation and the instrumented jar file must be added to the application CLASSPATH before any other JMS jar file.

Projects

Event collection projects enable you to easily group and manipulate communication links, data collection filters, database schemas, and Analyzers as one entity. An event collection project is used by an Analyzer to define the communication links, the data collection filters, and the database schema that data will be written into. A project is assigned to a single Analyzer host for event analysis.

Data Collection Filters

Data collection filters assigned to a project determine the amount and type of information collected by each Sensor. Data collection filters specify criteria such as the following:

- Which technologies, hosts, programs, or APIs to collect information about
- Which CICS regions, transactions, and job names to collect information about
- Which queues or queue managers to collect information about
- Which servlets, WebSphere applications, WebSphere servers and URIs to collect information about
- What time range to collect information for
- The level of detail to be collected, such as API name only, API name and call arguments, or API name, call arguments, and data buffer segment. For the Servlet Sensor, the default data buffer size is 1K.

Database Schemas

The project schema defines the tables into which the events collected by Sensors are stored. When the Analyzer retrieves and processes events collected by Sensors, it places them into event related tables. By using schemas to partition event data by project, you can control access to event data collected by each project.

TransactionVision Administration

This guide provides instructions for performing the following administration tasks:

- Planning the TransactionVision deployment (Chapter 2)
- Installing the TransactionVision web user interface, Analyzer, and Sensors (Chapter 3)
- Setup and configuration (Chapter 4)
- Managing user accounts (Chapter 5)
- Managing communication links (Chapter 5)
- Managing projects (Chapter 5)

- Configuring data collection filters (Chapter 5)

- Configuring the Analyzer (Chapter 5)

- Managing database schemas (Chapter 5)

- Extending TransactionVision Sensors (Chapter 7)

- Resolving TransactionVision problems (Chapter 6)

For information on using TransactionVision views and reports to identify and resolve potential problems in your transaction-based applications, see the *TransactionVision User's Guide*.

Starting TransactionVision for Administration

The TransactionVision web user interface runs in a web browser. To access it, start your browser and open the URL for the TransactionVision web user interface application. This URL has the following format:

http://*hostname*:*port*/*root_url*

Field	Description
hostname	The hostname of the application server.
port	The appserver port number used by the virtual host. For a default WebSphere setup using *default_host* as the virtual host, this is typically port 9080.
root_url	`/tv` or the value of the `root_url` entry in the `tvision-ws-config.properties` file (WebSphere 4 only). See the *TransactionVision Installation and Configuration Guide* for more information about this file.

For example, if your application server's hostname is *company_was* and the root URL for the TransactionVision web user interface application is */tv*, the URL for starting the TransactionVision web user interface is:

```
http://company_was:9080/tv/
```

Browser Security Settings

For the TransactionVision user interface to function properly, you must configure your browser security settings to enable it to run the Java and SVG plug-ins used by TransactionVision.

To verify or change browser security settings in IE6, perform the following steps:

1. In your browser window, choose Tools > Internet Options. The Internet Options dialog opens.

2. On the Security tab of the Internet Options dialog, click the icon for the zone for the server running TransactionVision and click Custom Level. The Security Settings dialog opens.
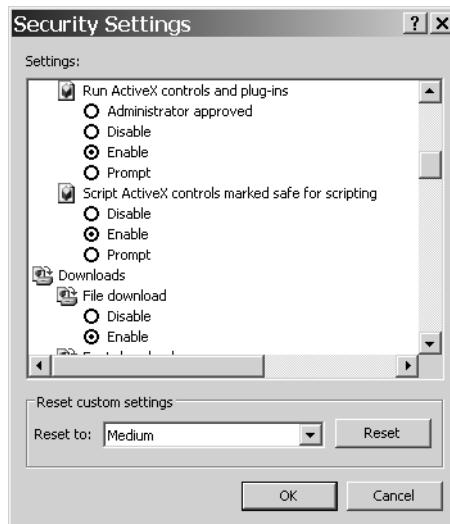
3. Scroll to the Run ActiveX controls and plug-ins setting and click the Enable radio button.

4. Click OK.

Important! If the server running TransactionVision is in the Internet zone, you may wish to add it to the Trusted Sites zones instead. By default, the Trusted Sites zone has all the necessary security settings for TransactionVision to function properly, and you do not have to enable ActiveX controls and plug-ins for the entire Internet zone.

Logging Into TransactionVision

When you enter the URL for the TransactionVision web user interface application, the TransactionVision login page appears.

BRISTOL TECHNOLOGY˙
Smart transactions. Smart move.

TransactionVision®

T O T A L   B U S I N E S S   V I S I B I L I T Y

User ID: [＿＿＿＿＿]
Password: [＿＿＿＿＿]
[Login]

5. Type your user name in the Login field. The user name must match an LDAP user entry. For more information about managing users, see Chapter 8, "Managing Users."

6. Press the Tab key and type your password in the Password field.

7. Click *Login* to display your TransactionVision home page.

Enabling Automatic Login

You can enable automatic login by storing login information in a cookie file. This automatic login method enables a "Remember Login?" checkbox on a the login page. If it is checked, TransactionVision saves the user name and password in a cookie file so that you are not prompted to enter this information the next time you login on the same computer. Instead, you are automatically logged in with the saved user name and password, and the TransactionVision home page is displayed. Because login information is saved in a cookie file, cookies must be enabled in your browser. Note that the password is encrypted in the cookie file using the Java JCE interface.

Important! When automatic login is enabled, login information is only remembered if you close the TransactionVision browser window without clicking *Logout*. If you log out explicitly by clicking this button, login information is **not** saved.

To enable automatic login with a cookie file, perform the following steps. Note that the first three steps are required **only** if JDK 1.3.1 or below is used. If you are using JDK 1.4 or above, start at Step 4.

1. Determine whether IBMJCE is installed as the security provider. Look for the following line in your WAS `<JAVA_HOME>/jre/lib/security/java.security` file:

   `security.provider.2=com.ibm.crypto.provider.IMBJCE`

   If this entry exists, proceed to **Step 4**. If IBMJCE is **not** installed, continue with **Step 2**.

2. Download JCE 1.2.1 from the following link:

   `http://java.sun.com/products/jce`

3. Unzip JCE 1.2.1 and copy `sunjce_provider.jar` to the WAS `<JAVA_HOME>/jre/lib/ext` directory. In the same directory, you may find `US_export_policy.jar` and `local_policy.jar`. If not, copy these files from JCE 1.2.1 as well.

4. Modify the following properties in the `<TVISION_HOME>/config/ui/UI.properties` file:

   - `autologin`
     Set this property to **true** to enable automatic login or **false** to disable it.

   - `jceProvider`
     Set this property to the name of the JCE provider library. The default is to use IBM JCE for encrypting/decrypting passwords. If you downloaded JCE 1.2.1, set this property as follows:

     `jceProvider=com.sun.crypto.provider.SunJCE`

5. Restart the TransactionVision web user interface using your application server administration console.

Setting Session Timeout

The session timeout can be set through the administration console of your application server. In Weblogic, you can change the Session Timeout setting under the Configuration > Descriptor tab of your deployed application. In WebSphere you can set it on the Web Container > Session Management tab of your application server, or individually for an application in its Session Management section. For more details refer to the appropriate WebSphere or Weblogic documentation.
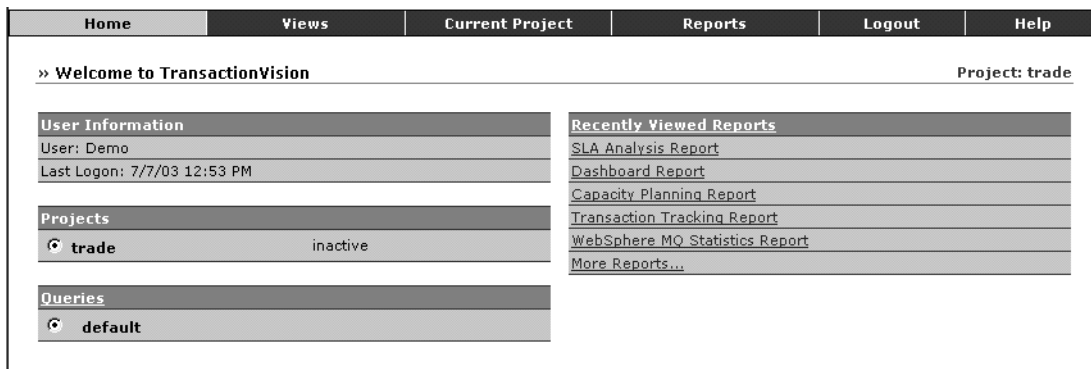
Multiple Sessions

When you open a new browser window from within a TransactionVision session (for example, by pressing <CTRL>+N), the new window is part of the same session. If you start a new instance of the browser and login to the TransactionVision web user interface, a new session is started. You may wish to use multiple sessions for reasons such as the following:

- You want to view and compare the results of two different queries for the same project. If you open a new window in the same session to do this, the new window reverts to the current query when it is refreshed.

- More than one user needs to login with the same user ID and password.

If you start an additional session from a different IP address than the current session, TransactionVision displays a warning that another user with the same user ID is logged in and that changes in your session and the other user's session may overwrite each other. To continue with the session, click Yes. To cancel and login with a different user ID, click No.

The TransactionVision Home Page

When you login to TransactionVision, the TransactionVision home page is displayed.

Home	Views	Current Project	Reports	Logout	Help

» Welcome to TransactionVision   Project: trade

User Information	**Recently Viewed Reports**
User: Demo	SLA Analysis Report
Last Logon: 7/7/03 12:53 PM	Dashboard Report
	Capacity Planning Report
Projects	Transaction Tracking Report
⊙ trade inactive	WebSphere MQ Statistics Report
	More Reports...
Queries	
⊙ default	

Menus

All TransactionVision user and administrator operations can be performed by selecting the appropriate menu item. If a user does not have access to a menu hierarchy, it will not appear in the menu bar. The following menus are available:

Menu	Description
Home	Displays your TransactionVision home page.

Menu	Description
Views	Provides access to the Component Topology Analysis, Transaction Analysis, and Event Analysis views. See the *TransactionVision User's Guide* for instructions on using these views.
Current Project	Provides project status information as well as information about scheduled jobs, communication links, data collection filters, and queries used by the project. For instructions on using queries, see the *TransactionVision User's Guide*. • For information about data collection filters, see Chapter 6, "Managing Data Collection Filters." • For information about communication links, see Chapter 5, "Managing Communication Links." • For more information about scheduled jobs, see Chapter 7, "Managing Scheduled Jobs."
Administration	Provides commands for managing projects, jobs, Analyzers, communication link templates, and database schemas. This menu is only visible if you have access rights to perform administrative operations. • For more information about user access rights, see Chapter 8, "Managing Users." • For more information about projects and database schemas, see Chapter 4, "Managing Projects." • For information about managing services, see Chapter 3, "Managing Analyzers." • For information about communication link templates, see Chapter 5, "Managing Communication Links." • For more information about jobs, see Chapter 7, "Managing Scheduled Jobs."
Reports	Displays the Reports page, which provides links to all reports for which you have the correct access rights. For instructions on using the standard reports installed with TransactionVision, see the *TransactionVision User's Guide*. For information about creating custom reports, see the *TransactionVision Programmer's Guide*.

Menu	Description
Logout	Ends your current TransactionVision session and displays the Login page.
Help	Provides access to online help, documentation, TransactionVision version information, and links to support information on Bristol's web site.

User Information

The User Information area lists the user name you are logged on as. It also lists the date and time of your last logon.

Projects

The Projects area lists all TransactionVision projects that you have access rights to use for analysis. For each project, the project name, status, and optional description is listed.

Click the radio button to the left of a project name to make it the active project for your session. When you display the Component Topology Analysis, Transaction Analysis, or Event Analysis views, the views are created with data from the active project.

Click the Projects link to display more detailed information about the active project.

Queries

The Queries area lists all available queries for the active project. For each query, the query name and optional description is listed.

Click the radio button to the left of a query name to make it the active query for your project. When you display the Component Topology Analysis, Transaction Analysis, or Event Analysis views, the views are created with data from the active project that meet the active query.

Click the Queries link to display more detailed information about available queries.

Recently Viewed Reports

The Recently Viewed Reports area lists TransactionVision the reports most recently viewed by the current user. Click a report name to access it. Click the Recently Viewed Reports link to display a list of all available and saved reports.

Using the TransactionVision Samples

To help you understand the types of transaction information you can track with TransactionVision, the TransactionVision installation provides two sample databases that you can import into TransactionVision:

- Trade is a sample based on a stock-trading system that processes a number of different classes of transactions.

- Account is a sample that has add, display, delete, and update commands to a CICS VSAM file through 3270 terminal input and query commands to the VSAM file from a transaction invoked by the WebSphere MQ-CICS bridge in response to WebSphere MQ messages on the bridge request queue.

Using the Trade Demo

TransactionVision provides a sample event database named Trade. You can use this demo to become familiar with the features and capabilities of TransactionVision.

The Trade demo database contains simulated statistics data spanning over two months. This simulated data is for demonstration purposes only.

The steps for importing the Trade demo data so that users can access it depend on whether you are using DB2 or Oracle as your DBMS.

Files Installed with the Trade Demo

File	Description
readme.txt	Readme file
trade_db2.zip	Zip file containing DB2 IXF import table archives.
trade_oracle.zip	Zip file containing Oracle table dump archive.
TradeTransaction.xdm	Trade demo XML database mapper transaction definition file.
TradeTransaction-Class.xdm	Trade demo XML database mapper transaction class definition file.
create_user_trade.sql	Oracle SQL script to generate the user TRADE.
create_project_trade.sql	Oracle SQL script to insert the TRADE schema into the system table.

File	Description
transactionclass.sql	Transaction class definition file for this demo (for reference only).

Importing Trade Demo Data for DB2:

Perform the following steps to import the trade data using DB2:

1. Copy all files from the `<cd root>/Demos/trade` directory to a location on your system's hard drive. Using a command prompt or terminal window, cd to this directory. All commands listed in subsequent steps should be run from this directory.

2. Unzip the demo database files from the `trade_db2.zip` archive. On Windows, use either the Windows Compressed Folder utility or WinZip. On UNIX platforms, use the command:

   ```
   unzip trade_db2.zip
   ```

3. Make sure that the database is ready for import (no active transactions).

4. Copy the following files to the `<TVISION_HOME>/config/xdm` directory:

   - `<TVISION_HOME>/demos/trade/TradeTransaction.xdm`

   - `<TVISION_HOME>/demos/trade/TradeTransactionClass.xdm`

5. Create the TRADE database schema tables. In the folder `<TVISION_HOME>/demos/trade/trade_ixf`, run the following command:

   ```
   $TVISION_HOME/bin/CreateSqlScript[.sh|.bat] -n -e -c -p TRADE
   ```

6. If you are using Windows, open a DB2 command window in either of the following ways:

   - Choose the Start>All Programs>IBM DB2>Command Line Tools>Command Window menu selection.

   - Run the command **db2cmd** from the Command Prompt.

   If you are using UNIX, make sure that your database environment is set up properly. This may require logging in under the appropriate database instance user ID. Contact your database administrator if assistance is required.

7. In a DB2 command window, connect to the TVISION database:

   ```
   db2 connect to TVISION
   ```

8. In a DB2 command window, run the following command to populate the transaction class table used by reports in the TRADE project:

   ```
   db2 -n -t -f $TVISION_HOME/demos/trade/transactionclass.sql
   ```

9. In the folder `<TVISION_HOME>/demos/trade/trade_ixf`, run the following command to generate the sql script to import the ixf files:

   ```
   $TVISION_HOME/bin/CreateSqlScript[.sh|.bat] -i -p TRADE
   ```

10. Remove the first two lines (shown below) from the `import_project_tables.sql` file:

    ```
    LOAD FROM RAW_EVENT.IXF OF IXF LOBS FROM .\ MODIFIED BY
    lobsinfile INSERT INTO TRADE.RAW_EVENT;
    ```

    ```
    RUNSTATS ON TABLE TRADE.RAW_EVENT AND INDEXES ALL;
    ```

11. Run the generated script with the following command to load the database with the Trade demo data:

    ```
    db2 -n -t -f import_project_tables.sql
    ```

Important! Note that the import operation could take a long time to complete!

12. Using the TransactionVision user interface, create a project called trade (lowercase) with the schema TRADE:

    • Choose Administration > Projects and click New Project...

    • Specify the project name **trade** and click Next>.

    • Use the existing database schema **TRADE**, uncheck Apply timeskew to collected events, and click Next>.

    • It is not necessary to assign any Analyzer. Click Next>.

    • Uncheck the deleteEvents and Transaction Statistics jobs, then click Next>.

    • Click Finish.

13. Using the TransactionVision user interface, run the reports on the Reports page or browse the views to see the data in this project.

14. The folder containing the trade demo files copied from the CD is no longer needed and can be deleted.

15. If you want to continue collection using the TRADE schema, run the follwoing command to reset the database sequence:

    ```
    CreateSqlScript[.sh|.bat] -r -p TRADE
    ```

Importing Trade Demo Data for Oracle:
Perform the following steps to import the trade data using Oracle:

1. Copy all files from the `<cd root>/Demos/trade` directory to a location on your system's hard drive. Using a command prompt or terminal window, cd to this directory. All commands listed in subsequent steps should be run from this directory.

2. Unzip the demo database files from the `trade_oracle.zip` archive. On Windows, use either the Windows Compressed Folder utility or WinZip. On UNIX platforms, use the command:

   ```
   unzip trade_oracle.zip
   ```

3. Make sure that the database is ready for import (no active transactions).

4. Copy the following files to the `<TVISION_HOME>/config/xdm` directory:

   - `<TVISION_HOME>/demos/trade/TradeTransaction.xdm`

   - `<TVISION_HOME>/demos/trade/TradeTransactionClass.xdm`

5. If you are using Windows, open a SQL Plus command window by choosing the Start>All Programs>Oracle>Application Development>SQL Plus menu selection.

   If you are using UNIX, make sure that your database environment is set up properly. This may require logging in under the appropriate database user ID. Contact your database administrator if assistance is required.

6. Create the user TRADE:

   ```
   sqlplus create_user_trade.sql
   ```

7. Run the Oracle imp utility to import TRADE data. The following is a sample command:

   ```
   imp trade/trade file=trade_oracle.dmp fromuser=trade
   touser=trade
   ```

8. Insert the TRADE schema into the system table with the following command:

   ```
   sqlplus create_project_trade.sql
   ```

9. Using the TransactionVision user interface, create a project called trade (lowercase) with the schema TRADE:

   - Choose Administration > Projects and click New Project...

   - Specify the project name **trade** and click Next>.

   - Use the existing database schema **TRADE**, uncheck Apply timeskew to collected events, and click Next>.

   - It is not necessary to assign any Analyzer. Click Next>.

   - Uncheck the deleteEvents and Transaction Statistics jobs, then click Next>.

- • Click Finish.

10. Using the TransactionVision user interface, run the reports on the Reports page or browse the views to see the data in this project.

11. If you want to continue collection using the TRADE schema, reset the database sequence:

```
<TVISION_HOME>/bin/CreateSqlScript[.sh|.bat] -n -e -d -p TRADE
```

12. The folder containing the trade demo files copied from the CD is no longer needed and can be deleted.

Trade Demo Cleanup

To drop the TRADE database schema, run the following command:

```
CreateSqlScript[.sh|.bat] -n -e -d -p TRADE
```

Using the Account Demo

In addition to the Trade demo, TransactionVision also provides a sample event database named Account.

Files Included with the Account Demo

File	Description
readme.txt	Readme file
account_db2.zip	Zip file containing DB2 IXF import table archives.
account_oracle.zip	Zip file containing Oracle table dump archive.
create_user_account.sql	Oracle SQL script to generate the user ACCOUNT.
create_project_account.sql	Oracle SQL script to insert the ACCOUNT schema into the system table.
transactionclass.sql	Transaction class definition file for this demo (for reference only).

The steps for importing the Account demo data so that users can access it depend on whether you are using DB2 or Oracle as your DBMS.

Importing Account Demo Data for DB2:

Perform the following steps to import the account data using DB2:

1. Copy all files from the `<cd root>/Demos/account` directory to a location on your system's hard drive. Using a command prompt or terminal window, cd to this directory. All commands listed in subsequent steps should be run from this directory.

2. Unzip the demo database files from the `account_db2.zip` archive. On Windows, use either the Windows Compressed Folder utility or WinZip. On UNIX platforms, use the command:

   ```
   unzip account_db2.zip
   ```

3. Make sure that the database is ready for import (no active transactions).

4. Create the ACCOUNT database schema tables. In the folder `<TVISION_HOME>/demos/account/account_db2`, run the following command:

   ```
   $TVISION_HOME/bin/CreateSqlScript[.sh|.bat] -n -e -c -p ACCOUNT
   ```

5. If you are using Windows, open a DB2 command window in either of the following ways:

   - Choose the Start>All Programs>IBM DB2>Command Line Tools>Command Window menu selection.

   - Run the command **db2cmd** from the Command Prompt.

   If you are using UNIX, make sure that your database environment is set up properly. This may require logging in under the appropriate database instance user ID. Contact your database administrator if assistance is required.

6. In a DB2 command window, connect to the TVISION database:

   ```
   db2 connect to TVISION
   ```

7. In the folder `<TVISION_HOME>/demos/account/accounting_db2`, run the following command to generate the sql script to import the ixf files:

   ```
   $TVISION_HOME/bin/CreateSqlScript[.sh|.bat] -i -noLob -p ACCOUNT
   ```

8. Run the generated script with the following command to load the database with the Account demo data:

   ```
   db2 -n -t -f import_project_tables.sql
   ```

Important! Note that the import operation could take a long time to complete!

9. Using the TransactionVision user interface, create a project called **account** (lowercase) with the schema **ACCOUNT**:

   - Choose Administration > Projects and click New Project...

   - Specify the project name **account** and click Next>.

- Use the existing database schema **ACCOUNT**, uncheck Apply timeskew to collected events, and click Next>.

- It is not necessary to assign any Analyzer. Click Next>.

- Uncheck the deleteEvents job, check the Transaction Statistics job, and click Next>.

- Click Finish.

10. Using the TransactionVision user interface, run the reports on the Reports page or browse the views to see the data in this project.

11. The folder containing the Account Demo files copied from the CD is no longer needed and can be deleted.

12. If you want to continue collection using the ACCOUNT schema, run the follwoing command to reset the database sequence:

```
CreateSqlScript[.sh|.bat] -r -p ACCOUNT
```

Importing Account Demo Data for Oracle:

Perform the following steps to import the account data using Oracle:

1. Copy all files from the `<cd root>/Demos/account` directory to a location on your system's hard drive. Using a command prompt or terminal window, cd to this directory. All commands listed in subsequent steps should be run from this directory.

2. Unzip the demo database files from the `account_oracle.zip` archive. On Windows, use either the Windows Compressed Folder utility or WinZip. On UNIX platforms, use the command:

```
unzip account_oracle.zip
```

3. Make sure that the database is ready for import (no active transactions).

4. If you are using Windows, open a SQL Plus command window by choosing the Start>All Programs>Oracle>Application Development>SQL Plus menu selection.

   If you are using UNIX, make sure that your database environment is set up properly. This may require logging in under the appropriate database user ID. Contact your database administrator if assistance is required.

5. Create the user ACCOUNT:

```
sqlplus create_user_account.sql
```

6. Run the Oracle imp utility to import ACCOUNT data. The following is a sample command:

```
imp account/account file=account_oracle.dmp fromuser=account
touser=account
```

7. Insert the ACCOUNT schema into the system table with the following command:

```
sqlplus create_project_account.sql
```

8. Using the TransactionVision user interface, create a project called account (lower-case) with the schema ACCOUNT:

   • Choose Administration > Projects and click New Project...

   • Specify the project name **account** and click Next>.

   • Use the existing database schema **ACCOUNT**, uncheck Apply timeskew to collected events, and click Next>.

   • It is not necessary to assign any Analyzer. Click Next>.

   • Uncheck the deleteEvents and Transaction Statistics jobs, and click Next>.

   • Click Finish.

9. Using the TransactionVision user interface, run the reports on the Reports page or browse the views to see the data in this project.

10. If you want to continue collection using the ACCOUNT schema, reset the database sequence:

```
<TVISION_HOME>/bin/CreateSqlScript[.sh|.bat] -n -e -d -p ACCOUNT
```

11. The folder containing the Account Demo files copied from the CD is no longer needed and can be deleted.

Account Demo Cleanup

To drop the ACCOUNT database schema, run the following command:

```
CreateSqlScript[.sh|.bat] -n -e -d -p ACCOUNT
```

Contacting Bristol Support

If you encounter a problem using 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 (US & Far East)
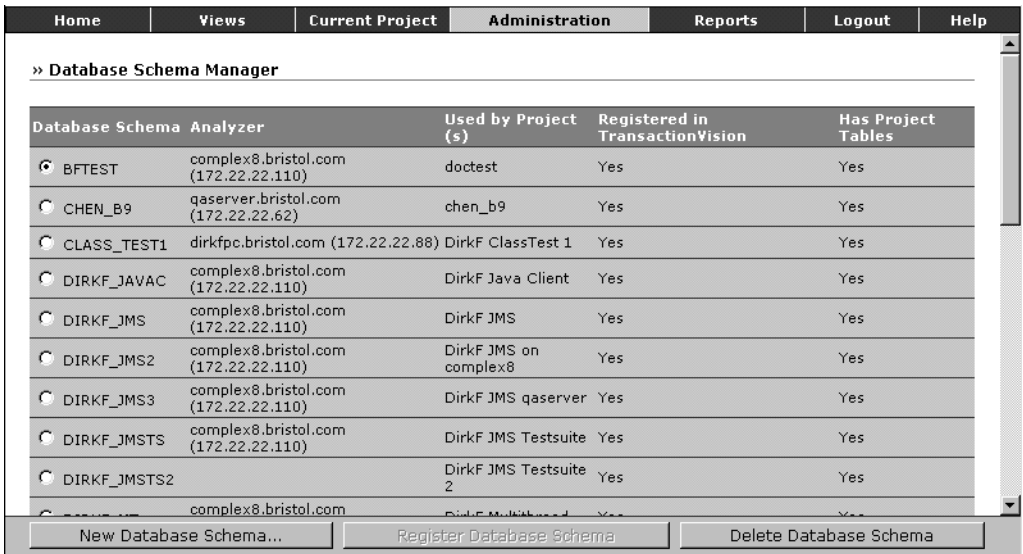 or **+31-33-450-5050** (Europe, Africa, and Middle East)

Managing Database Schemas

When the Analyzer processes event data collected by Sensors, it stores the results in database schemas assigned to TransactionVision projects.

Only users with access to the Administration menu may manage database schemas. See Chapter 8, "Managing Users," for information about permissions.

View All Database Schemas

To display the Database Schema Manager, choose the Administration > Database Schemas menu item.

Home	Views	Current Project	Administration	Reports	Logout	Help

» Database Schema Manager

Database Schema	Analyzer	Used by Project (s)	Registered in TransactionVision	Has Project Tables
⊙ BFTEST	complex8.bristol.com (172.22.22.110)	doctest	Yes	Yes
○ CHEN_B9	qaserver.bristol.com (172.22.22.62)	chen_b9	Yes	Yes
○ CLASS_TEST1	dirkfpc.bristol.com (172.22.22.88)	DirkF ClassTest 1	Yes	Yes
○ DIRKF_JAVAC	complex8.bristol.com (172.22.22.110)	DirkF Java Client	Yes	Yes
○ DIRKF_JMS	complex8.bristol.com (172.22.22.110)	DirkF JMS	Yes	Yes
○ DIRKF_JMS2	complex8.bristol.com (172.22.22.110)	DirkF JMS on complex8	Yes	Yes
○ DIRKF_JMS3	complex8.bristol.com (172.22.22.110)	DirkF JMS qaserver	Yes	Yes
○ DIRKF_JMSTS	complex8.bristol.com (172.22.22.110)	DirkF JMS Testsuite	Yes	Yes
○ DIRKF_JMSTS2		DirkF JMS Testsuite 2	Yes	Yes
○	complex8.bristol.com	DirkF Multithread	Yes	Yes

New Database Schema...	Register Database Schema	Delete Database Schema

This page provides the following information for each schema available in your environment:

- The schema name

- The host name and IP address of the Analyzer host that is assigned to the schema. Analyzer hosts are assigned to the schema by the Project wizard or the Update Analyzer page.

- The names of any projects the schema is assigned to. Schemas are assigned to projects by the Project wizard.

- Whether the schema has been registered in TransactionVision. A schema is registered in TransactionVision when any project uses the schema. However, if all projects using the schema are deleted, it is still registered in TransactionVision.

- Whether TransactionVision project tables have been created.

In general, when a project is created using the TransactionVision web user interface, the "Registered in TransactionVision" and "Has Project Tables" columns will have the value Yes. However, if a database administrator creates or manages schemas/tables using database tools (instead of the TransactionVision web user interface), a scenario may occur where the project tables exist, but not the corresponding entries in the administration tables of the TransactionVision web user interface.

Similarly, a schema and its tables might be deleted using external database tools, but a project might still use the schema. In that case, this page provides the status of these schemas in relation to TransactionVision projects and allows you to fix any errors.

New Database Schema...

To create a new schema, click New Database Schema.... The Create a New Database Schema page appears. You may also create a new schema when you create a project with the Project wizard. See "Create a New Database Schema" for information on creating a schema.

Register Database Schema

To register an unregistered schema, click the radio button next to the schema name to select it and click Register Database Schema.

Create Tables

To create tables for a schema that does not have project tables, click the radio button next to the schema name to select it and click Create Tables.

Delete Database Schema

To delete a schema, click the radio button next to the schema name to select it and click Delete Database Schema.

Important! Deleting a database schema permanently erases all event data in projects associated with the database schema. This data is not recoverable, so use this command with caution. You may not delete a schema if an Analyzer is assigned to it.

You will be prompted to confirm the deletion; click Yes to proceed.

Create a New Database Schema

Each project must be assigned one database schema. To create a new database schema, click New Database Schema... on the Database Schema Manager page.

| Home | Views | Current Project | Administration | Reports | Logout | Help |

» Create A New Database Schema

Enter new database schema name

Name: Test_Schema

| Finish | | Cancel |

Enter the name of the new database schema and click Finish. The schema name may contain only alphanumeric characters and the underscore character. It must begin with an alphabetical character.

The new database schema is registered in TransactionVision and all TransactionVision related tables are created.

Managing Analyzers

TransactionVision uses the Analyzer service to perform the following tasks:

- Send configuration messages to Sensors and collect event messages from Sensors
- Process events from the Sensor, correlate them into transactions, and store them in the project schema.

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

Only users with required permissions may manage services. See Chapter 8, "Managing Users," for information about permissions.

View All Analyzer Hosts

An Analyzer host is a host running the Analyzer service. To view the Analyzers page, choose the Administration > Analyzers menu item.
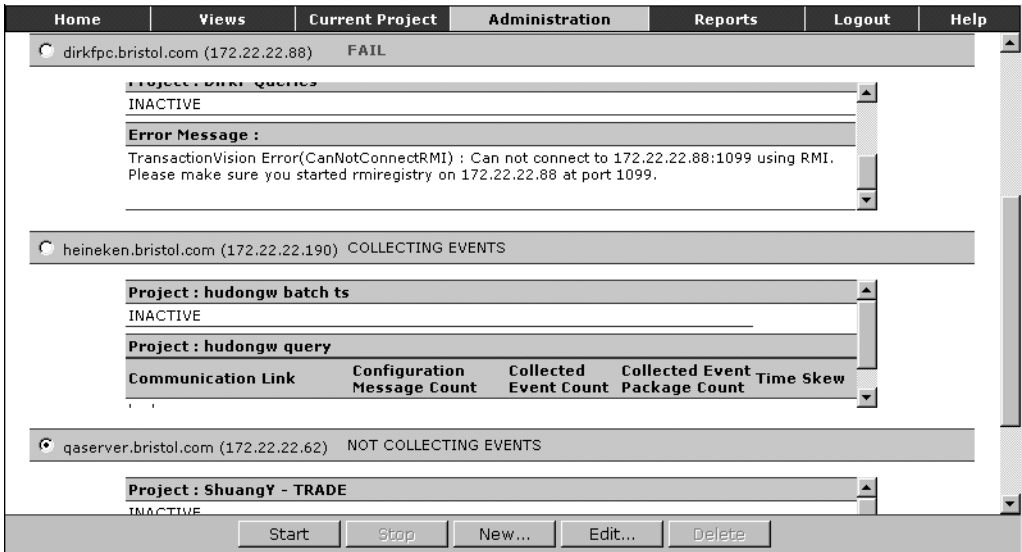
This page displays the following information about each Analyzer available in your environment:

- The host name and IP address of the host on which the TransactionVision Analyzer component is installed.

- The Analyzer status. The following table describes possible status values:

Status	Description
COLLECTING EVENTS	The Analyzer is started and collecting events.
FAIL	TransactionVision cannot connect to the Analyzer; an error message is provided.
RECOVERY	The Analyzer is performing statistics recovery. This can happen when the Analyzer has been stopped abnormally (for example, the process has been killed), and not all event statistics have been flushed to disk. In this case the Analyzer will reprocess the last events to get the statistics up-to-date before it starts the normal processing.

Status	Description
QUIESCING	The Analyzer is shutting down all collection threads in response to a stop command. Once all collection threads are shut down, the status changes to NOT COLLECTING EVENTS.
NOT COLLECTING EVENTS	The Analyzer has been stopped and is not collecting events.

• The name and status of each project that the Analyzer is assigned to.

• If the status is COLLECTING EVENTS, the following details for each communication link for each active project:

  • Communication link name

  • Number of configuration messages sent since the Analyzer was started or refreshed

  • Number of events collected on the communication link

  • Number of event packages sent on the communication link

  • Amount of time skew across the communication link

Start/Stop

To begin collecting events on an Analyzer host, click the radio button next to the host name to select it and click Start. The Analyzer connects to assigned communication links and starts collecting events from the event queues. The status changes to COLLECTING EVENTS and details about event collection are displayed for each communication link for each active project.

If the RECOVERY status is displayed, the Analyzer is performing statistics recovery. This can happen when the Analyzer has been stopped abnormally (for example, the process has been killed), and not all event statistics have been flushed to disk. In this case the Analyzer will reprocess the last events to get the statistics up-to-date before it starts the normal processing.

To stop collecting events from communication link event queues on an Analyzer host, click the radio button next to the host name to select it and click Stop. The Analyzer shuts down all collection threads without waiting for them to clean out events in the event queue. While the Analyzer is shutting down all the threads, its status is set to QUIESCING. Once all collection threads are shut down, the status changes to NOT COLLECTING EVENTS.

Important!   If an Analyzer ends abnormally, there may be XML events in the database that have not been processed for transaction analysis yet. When the Analyzer is started, it scans the event table for unprocessed events and recovers the statistics cache. To recover events for analysis, put failed events from the dead letter queue back to the event queue with the `GetPut` utility. For more information about this utility, see "GetPut" on page 188.

New...

To add a new Analyzer host to your environment, click New.... The Create Analyzer page appears. See "Create or Edit an Analyzer Host" for information on adding a new Analyzer host.

Edit...

To change the Analyzer configuration settings for a host, click the radio button next to the host name to select it and click Edit.... The Update Analyzer page appears. See "Create or Edit an Analyzer Host" for information on modifying settings for an Analyzer host. If the Analyzer is collecting events, the Edit... button is disabled; click Stop to stop the Analyzer to edit it.

Delete

To remove an Analyzer host from your environment, click the radio button next to the host name to select it and click Delete. You may only delete an Analyzer host if no projects are assigned to it. You will be prompted to confirm the deletion; click Yes to proceed.

Create or Edit an Analyzer Host

Create an Analyzer host to identify a host running the Analyzer service. You must create an Analyzer host to assign communication links for a TransactionVision project to it. Once an Analyzer host is created, you may modify settings.

You may access the Create Analyzer page from the Analyzers page or Project wizard. You may access the Update Analyzer page from the Analyzers page.

To create or update an Analyzer host, complete the following fields and click Finish. If applicable, the valid value range is shown to the right of each field.

Field	Description
Analyzer Host	Enter the name of the host to add (for example, host1.bristol.com). The Analyzer service component must be installed on this host. Only one instance of the Analyzer may run on a given host; however, it can handle multiple projects.
Connection Retry Delay	Specify the number of seconds to wait between attempts to connect to the queue manager for the communication link event queue. The default is 10 seconds.

Field	Description
Database Exception Threshold	Specify the number of exceptions from the database to accept in the event collection threads before attempting to reconnect to the database.
Database Reconnect Interval	Specify the time in seconds to wait between tries to connect to the database. This value is used if the first attempt to connect to the database fails. The default is 10 seconds.
Event Get Wait Interval	Specify the number of seconds to wait for an event message. Increasing this number slows down the Analyzer response to commands. The default is 5 seconds.
Time Skew Reply Wait Interval	Specify the number of seconds to wait for the message sent to determine the time skew across a communication link. The default is 30 seconds. If no response is received in this time, the time skew is disregarded.
Time Skew Check Interval	Specify the number of seconds between sending messages to check determine the time skew across the communication link. The default is 3600 seconds (one hour). At each specified interval, the Analyzer takes time skew samplings across the communication links and, if it is running on a different server, the TransactionVision application server.
Time Skew Latency Threshold	Specify the amount of time in milliseconds to accept when sampling time skews. The default value is 100 milliseconds. If more than the specified time passes when waiting for a time skew reply, another time skew sampling is taken in hopes it will have a lower latency. Any time skew samples that surpass the threshold are still kept in the time skew history because it is possible no samplings will ever be under the suggested latency threshold. This threshold affects time skew sampling across communication links and to the application server.

Field	Description
Time Skew Retry Threshold	Specify the maximum number of time skew samples to take at a single time skew check interval. At each time skew check interval, TransactionVision will continue to retry up to this number of times if the sampling surpasses the time skew latency threshold. The default number of retries is 8.
Time Skew History Size	Specify the number of previous time skew samplings to consider for the best time skew to use. The default is 24. Having a larger history size improves the chance of finding a time skew with smaller latency, but it also increases the chance of picking a time skew that may have occurred long ago to represent a time skew that may be more off due to clock drifting. A history size of 0 causes the time skew that was taken with the smallest latency to be used. This time skew is not discarded until a new time skew with equal or less latency is taken. Using this value disregards clock drift, which occurs frequently.
Time Skew Message Priority	Specify the WebSphere MQ Message Priority on the Confirmation of Arrival (COA) time skew messages. The default value is -1, which causes the default priority to be used. Since configuration and event messages use the default priority, it is a good idea to set this value at a value higher than the default. Setting a higher value ensures that time skew messages travel through channels faster than other messages that may be backed up on transmission queues. Therefore, latency times when sampling time skews is reduced and results are more accurate. This value must not exceed the MaxPriority attributes of the queue managers.

Field	Description
Configuration Message Expiry	Specify the expiration time in minutes of configuration messages sent from the Analyzer to the Sensors. When this value is set, configuration messages are sent two minutes before the old configuration message will expire. Setting an expiration time limits how long Sensors will continue to report events if the Analyzer is stopped in a non-proper fashion. When the configuration messages expire, Sensors stop reporting events. Set the Configuration Message Expiry 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. Specify value of -1 if you do **not** want configuration messages to expire. The Analyzer only sends configuration messages if filter criteria change, it has been told to start or stop, or a communication link is changed. For guidance on setting this value, see "Configuration Message Expiry" following this table.
Client Time Skew Servlet URL	Specify the URL of the servlet used by the Analyzer to determine the time skew between the Analyzer host and the host running the TransactionVision application server. This servlet is installed at the default URL when the TransactionVision web user interface is installed.
Analyzer port	Enter the port number where the Analyzer listens for RMI calls. The default value is 1099. If this system is changed on the server so that the Analyzer listens for RMI calls on another port, you must specify the correct port number.
Database Schema	Check the box for each database schema you wish to assign to the Analyzer. The Analyzer will be automatically assigned to all projects that use selected database schemas. You may only select project schemas that are not currently assigned to an Analyzer host. This field is available when editing an Analyzer, but not when creating a new Analyzer.

Configuration Message Expiry

The Analyzer depends on a database connection in order to maintain its configuration messages, which tell Sensors wether to collect events or not. When the database used by TransactionVision is shutdown, particularly for extended periods of time, configuration message expiry settings can affect how your Analyzer and Sensors perform. There are two ways the configuration message expiry can effect your collection of events:

- The configuration message expiry controls how long Sensors remain active after the database shuts down. If the database is down, the Analyzer will not be able to generate new configuration messages; after the time specified in the configuration message expiry settings, Sensors will automatically be disabled.

- The configuration message expiry setting also controls how often the Analyzer will attempt to re-establish communication and send out a new configuration message. It will be at most *configuration message expiry setting* minutes after the database is restarted that Sensors will be re-enabled.

Reasons for a High Configuration Message Expiry

If it is critical that your Sensors do not miss collecting any events, even those that occur while the database is down, you would want to set your configuration expiry to a high value. This means that Sensors would continue to collect for this long after the Analyzer Goes Inactive, And Place Their Messages In The Message Queue. When The Analyzer finally reconnects, it will process all the backed up messages on the queues.

Important! If the database were to be down for a long time, events would keep being backed up on the queues since they would never be processed. Make sure you have the necessary system resources set to be able to handle the backed up events if you have a high event volume, expect the database to be down for extended periods, and want to ensure that all events are saved during this period.

Reasons for a Low Configuration Message Expiry

On the other hand, if keeping all generated events during database downtime is not important, you could set your configuration expiry to a shorter value. In this way, Sensors will stop collecting after the specified time period, and be disabled until the Analyzer is able to reconnect and send out a new message. This would prevent unnecessary events from backing up while the analyzer is unable to process them.

To force the Analyzer to immediately resend a configuration message after the database has been shutdown, simply stop and restart the project.

Configure the Analyzer

Analyzer configuration information is stored in the `<TVISION_HOME>/config/ services/Analyzer.properties` file. For more information about this file, see Appendix C in the *TransactionVision Installation and Configuration Guide*.

Stopping and Restarting the Analyzer Process

To completely stop the Analyzer process on Windows, you can either run the **ServicesManager** script with the -exit flag, or you can stop the Analyzer through the standard Windows Services administration control panel. On UNIX, you must use the **ServicesManager** script with the -exit flag.

Important! There is also a -killserver flag to stop the Analyzer; however, it is not the recommended way of shutting down the Analyzer and should only be used if you need to quickly abort the Analyzer and the -exit option did not work. Using the - killserver flag will cause the Analyzer to exit without writing out its cache to disk, causing it to enter recovery mode upon restarting.

To restart the Analyzer, simply use the **ServicesManager** script with the -start option. On Windows, you can also start it with the Service control panel. When the Analyzer is started, it will automatically continue with collection on any projects that are in the active state.

Chapter 4
Managing Projects

Event data collected by TransactionVision for analysis is organized into projects. When you are logged into TransactionVision, one project is the *current project*—any project-related changes are applied to the current project, and any analysis views use data from the current project. Use commands under the Current Project menu to view the status for or make changes to the current project. Use commands under the Administration menu to create new projects or modify any existing projects.

View Current Project Status

The Project Status page displays information about the current project. Use this page to start and stop data collection or edit project settings.

To display the Project Status page, choose the Current Project > Project Status menu item. Note that if your project contains a large number of events (more than 5 million), this page can take a few moments to display.

The Project Status page provides the following information about the project:

Information	Description
Project	The name of the current project. To change the current project, click Home and select a new project on the TransactionVision Home page.
Database Schema	The name of the database schema associated with the project.
Status	Indicates whether the project is currently collecting events. If the status is ACTIVE, the project is collecting events according to the settings in the data collection filter. If the status is INACTIVE, the project is not currently collecting events.
Events in Database Schema	The number of events in the project database schema. Transaction-Vision calculates the event count he first time the Project Status page in displayed in a session. When the page is refreshed, the event count is **not** updated for performance reasons. To update the event count, click Update. Updating the event count does not refresh the other information on the Project Status page. Move your cursor over the Update button to view a tooltip that shows the date and time the event count was calculated.

Information	Description
Analyzer	The name, IP address, and status of the Analyzer host assigned to the schema associated with the project. Status values are:

<div style="margin-left: 2em;">

- COLLECTING EVENTS
- NOT COLLECTING EVENTS
- QUIESCING
- FAIL
- RECOVERY

</div>

If the project receives a stop command and its Analyzer is still cleaning out events in the event queue, the Analyzer status is set to QUIESCING and the Start and Stop buttons for the project are disabled. A FAIL status indicates that TransactionVision cannot connect to the Analyzer host. The RECOVERY status indicates that the Analyzer is performing statistics recovery. This can happen when the Analyzer has been stopped abnormally (for example, the process has been killed), and not all event statistics have been flushed to disk. In this case the Analyzer will reprocess the last events to get the statistics up-to-date before it starts the normal processing.

This page also displays a list of the communication links assigned to the Analyzer host. For each communication link, it shows:

- The number of configuration messages sent
- The number of events processed
- The number of unprocessed events on the event queue
- The number of event packages sent
- The time skew
- The data collection filters assigned to the communication link.

To view or edit a communication link, click the communication link name to display the Edit Project Communication Link page. See "Edit a Communication Link or Template" on page 76 for more information.

To view or edit an assigned data collection filter, click the filter name to open the Edit Data Collection Filter page. See "Edit a Data Collection Filter" on page 89 for more information.

Start/Stop

To activate the current project, click Start. To stop data collection, click Stop. Only users with required permissions may start or stop data collection. See Chapter 8, "Managing Users," for information about permissions.

Edit Project...

To change any of the following, click Edit Project... to display the Edit Project page:

- Description
- Whether to apply timeskew to collected events
- Communication link assignments
- Data collection filter assignments

See "Edit a Project" on page 51 for more information.

Refresh

To update the view to reflect any changes since it was created, click Refresh. When you refresh the page, the event count is **not** updated; click Update (to the right of the event count) to update the number of events in the project.

View All Projects

To display the project list, choose the Administration > Projects menu item.

The project list page provides the following information for all projects:

Information	Description
Project Name	The name of the project.
Database Schema	The name of the schema associated with the project.
Status	Indicates whether the project is currently collecting events. If the status is ACTIVE, the project is collecting events according to the settings in the data collection filter. If the status is INACTIVE, the project is not currently collecting events.

It also provides more detailed information about the current project:

Information	Description
Events in Database Schema	The number of events processed by the schema associated with the project.
Analyzer	The name, IP address, and status of the Analyzer host assigned to the schema associated with the project. Status values are:
	• COLLECTING EVENTS
	• NOT COLLECTING EVENTS
	• QUIESCING
	• FAIL
	• RECOVERY
	If the project receives a stop command and its Analyzer is still cleaning out events in the event queue, the Analyzer status is set to QUIESCING and the Start and Stop buttons for the project are disabled. A FAIL status indicates that TransactionVision cannot connect to the Analyzer host.
	This page also displays a list of the communication links assigned to the Analyzer host. For each communication link, it shows:
	• The number of configuration messages sent
	• The number of events processed
	• The number of unprocessed events on the event queue
	• The number of event packages sent
	• The time skew
	• The data collection filters assigned to the communication link.
	To view or edit a communication link, click the communication link name to display the Edit Project Communication Link page. See "Edit a Communication Link or Template" on page 76 for more information.
	To view or edit an assigned data collection filter, click the filter name to open the Edit Data Collection Filter page. See "Edit a Data Collection Filter" on page 89 for more information.

To select a different project as the current project, click in the circle to the left of the project name.

Start/Stop

> To activate the current project, click Start. To stop data collection, click Stop. Only users with required permissions may start or stop data collection. See Chapter 8, "Managing Users," for information about permissions.

New Project...

> To create a new project, click New Project... to start the Project Wizard. See "Create a New Project" on page 45 for more information.

Edit Project...

> To change any of the following, click Edit Project... to display the Edit Project page:
>
> - General project description
> - Analyzer host
> - Communication link assignments
> - Data collection filter assignments
>
> See "Edit a Project" on page 51for more information.

Delete Project

> To delete the current project, click Delete Project. This button is only enabled if the project is inactive. You will be prompted to confirm the deletion; click Yes to delete the project. Only users with required permissions may delete projects. See Chapter 8, "Managing Users," for information about permissions.

> *Important!* Deleting a project does **not** drop the database schema that is associated with the project.

Refresh

> To update the view to reflect any changes since it was created, click Refresh.

Create a New Project

> The Project Wizard guides you through the steps of creating a new project. To start the Project Wizard, choose the Administration > Projects menu item to display the Projects page and click New Project....

> *Important!* New projects may only be created by users with the necessary permissions. See Chapter 8, "Managing Users," for information about permissions.

Step 1: Project Name and Description

Enter the name and a brief description for your project. This information appears on the TransactionVision home, Projects, and Project Status pages.

Home	Views	Current Project	Administration	Reports	Logout	Help

» **Step 1: Project Name and Description** Project Wizard

Project Name and Description

Project Name: | My Project |

Project Description: | Test Project |

Click Next> to assign a schema to the project.

Step 2: Database Schema Setup

Each project must be assigned a database schema where the Analyzer stores events collected by Sensors.

Home	Views	Current Project	Administration	Reports	Logout	Help

» **Step 2: Database Schema Setup** Project Wizard
Project: My Project

Assign Database Schema: MYSCHEMA

⊙ Create New Database Schema: | MYSCHEMA |
○ Use Existing Database Schema: | TRADE ▼ |
☑ Apply timeskew to collected events

You may create a new schema for the project or use an existing schema. To create a new schema, enter the schema name. Note that schema names must begin with a letter and may contain only letters, numbers, and underscore characters.

To use an existing schema, click the down arrow and select a schema from the list. Note that users of projects using the same schema will see all events from all projects that the schema is assigned to. Creating separate schemas for different projects enables you to control access to event data.
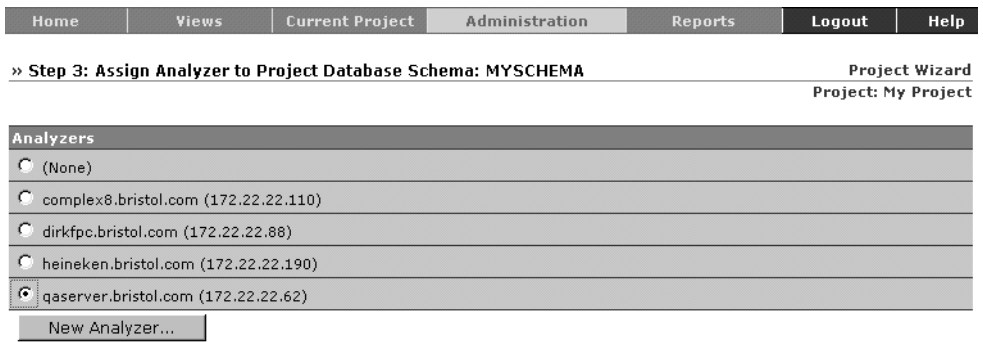
To apply the timeskew detected by TransactionVision on the communication link to vents, check the Apply timeskew to collected events checkbox. This box is checked by default. Note that disabling the timeskew calculation affects all projects that use the database schema assigned to your project.

Click Next> to assign an Analyzer host to the project.

Click <Back to return to the Product Description step or Cancel to exit the Project wizard.

Step 3: Assign Analyzer Host to Project Database Schema

Each project must be assigned an Analyzer host for correlating and performing transaction analysis on events in the schema.

Home	Views	Current Project	Administration	Reports	Logout	Help

» Step 3: Assign Analyzer to Project Database Schema: MYSCHEMA Project Wizard
Project: My Project

Analyzers
- ○ (None)
- ○ complex8.bristol.com (172.22.22.110)
- ○ dirkfpc.bristol.com (172.22.22.88)
- ○ heineken.bristol.com (172.22.22.190)
- ● qaserver.bristol.com (172.22.22.62)

New Analyzer...

A single Analyzer host may serve multiple schemas, but each schema may be served by only one Analyzer host. Therefore, if you assigned an existing schema to your project, the Analyzer host used by the schema is assigned to your project automatically.

If you created a new schema for your project, select an existing Analyzer host from the list or click New Analyzer... to define a new Analyzer host and select it. See "Create or Edit an Analyzer Host" on page 32 for more information on defining a new Analyzer host.

Click Next> to assign communication links to the Analyzer.

Click <Back to return to the Database Schema Setup step or Cancel to exit the Project wizard.

Step 4: Assign Communication Links to the Analyzer

The Analyzer used by a project must be assigned one or more communication links to allow a Sensor to communicate from a host on which an application is being monitored to the Analyzer. This step only occurs if you select an Analyzer in the previous step.
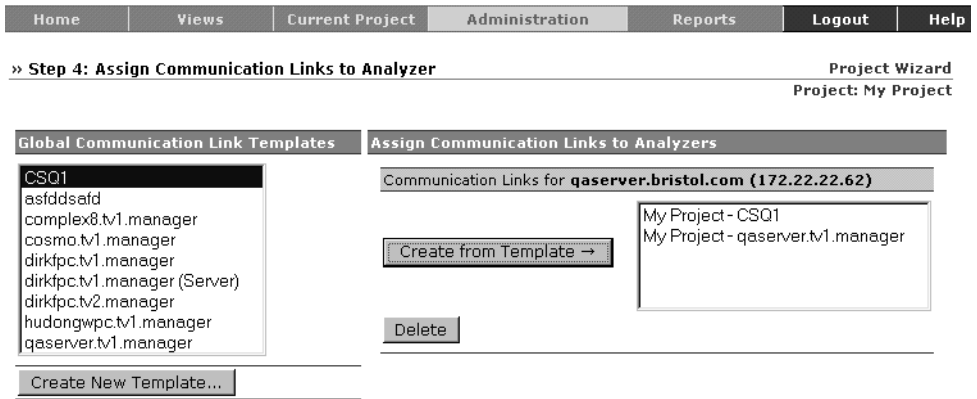
To assign a communication link to the Analyzer, select a communication link template in the list of global communication link templates and click Create from Template>. The communication link created for your project is a copy of the selected template; changes to the template do not affect the communication link used by your project.
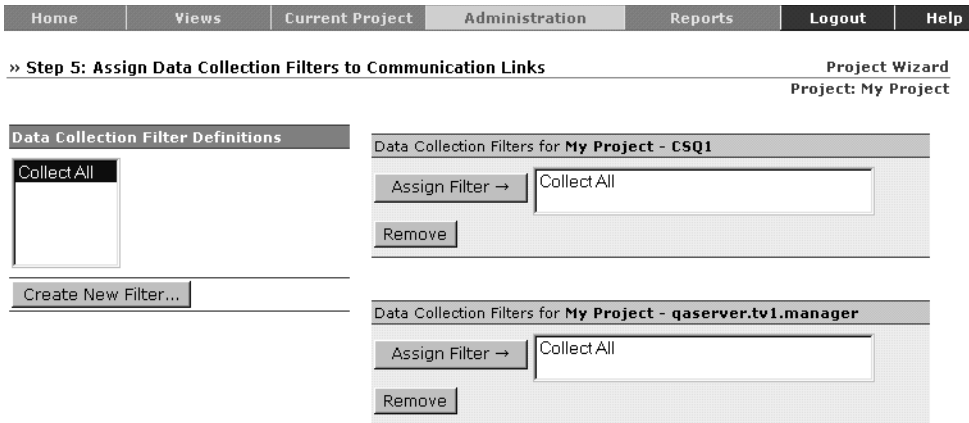
- To add a new global communication link template, click Create New Template... to start the Communication Link wizard. See "Create a New Communication Link or Template" on page 64 for information on creating communication link templates.

- To delete a communication link assigned to the Analyzer, select it in the list of communication links and click Delete.

Click Next> to assign data collection filters to communication links in your project.

Click <Back to return to the Assign Analyzer Host to Project Database Schema page or Cancel to exit the Project wizard.

Step 5: Assign Data Collection Filters to Communication Links

You must assign one or more data collection filters to the communication links for the Analyzer assigned to your project. By default the Collect All filter, which collects all events, is assigned. This step only occurs if you assign a communication link in the previous step.

| Home | Views | Current Project | Administration | Reports | Logout | Help |

» **Step 5: Assign Data Collection Filters to Communication Links**   **Project Wizard**
                                                                              Project: My Project

Data Collection Filter Definitions

Collect All

Create New Filter...

Data Collection Filters for My Project - CSQ1

Assign Filter →   Collect All

Remove

Data Collection Filters for My Project - qaserver.tv1.manager

Assign Filter →   Collect All

Remove

To assign a filter to the communication links for the Analyzer associated with your project, select the filter name in the Data Collection Filter Definitions list and click Assign Filter>. The filter name appears in the Data Collection Filters list.

To create a new data collection filter definition, click Create New Filter... to display the Add Filter page. See "Create a New Data Collection Filter" on page 86 for information on creating a filter.

You may assign any number of filters. Events that match **any** filter are collected by Sensors and forwarded to the Analyzer. If you create a new filter with specific filter conditions, make sure you remove the Collect All filter. Otherwise, all events will continue to be collected because they match the conditions of a filter. To remove a data collection filter from the list, select it and click Remove.

Click Next> to create jobs for your new project.

Click <Back to return to the Assign Communication Links to the Analyzer page or Cancel to exit the Project wizard.

Step 6: Create Jobs from Template

A job is a task that runs at a specified frequency. You may assign one or more jobs from available job templates to your project. A job instance is defined for this particular project; that instance can only be run on this project. However, multiple instances of the job bean may be run concurrently, each configured for a different project. For more information about jobs, see Chapter 7, "Managing Scheduled Jobs."

Important!　If the project you are creating shares a schema with another project, be aware that it will also share jobs with that project. For example, if you create a delete events job in project A and one in project B, if both projects share the same schema, the effects of both delete event jobs will be seen in both projects. When creating jobs for projects that share a schema, keep this in mind to prevent one project's jobs from having unintended consequences on data that is shared between the projects.

Home	Views	Current Project	Administration	Reports	Logout	Help

» **Step 6: Create Jobs from Template**　　　　　　　　　　　　　　Project Wizard
Project: My Project

Job Name	Job Description	Job Startup Mode	Job Interval	Job Class
☑ deleteEvents	Delete events that are older than 48 hours	manual	1 hours	com.bristol.tvision.job.DeleteEventsJobBean
☑ Transaction Statistics	Calculate business transaction statistics	automatic	10 minutes	com.bristol.tvision.job.TransactionStatisticsJob

Check the checkbox for any job you wish to create an instance of for your new project.

- The deleteEvents job deletes all events more than 48 hours old from your project database. This job is optional, and by default is turned off. To enable this job, go to the Job Status page after you finish creating the project.

- The Transaction Statistics job calculates business transaction statistics.This job is required if you wish to collect transaction statistics; these statistics are used by a number of different reports. If this job is not running, those reports will not be able to show any data.

To change the startup mode, job interval, or startup settings for jobs after your project is created, choose Current Project > Job Status to display the Job Status page. For more information, see "View Jobs for the Current Project" on page 130. For the deleteEvents job, in particular, you may wish to make the following changes to the startup options:

- By default, the value for the `-commit` option for the deleteEvents job is 100, meaning that the commit is done after the deletion of every 100 events. If you are deleting a large number of events, explicitly specify the parameter `-commit n` to commit after *n* event deletions. Otherwise, the database transaction log may overflow or the table may be locked for a long period of time.

- By default, the deleteEvents job and script do not allow deleting events in the last 48 hours. To override the default of 48 hours, use the `-minage` number-of-hours option. For example, `-minage 10` prevents the job from deleting events from the last 10 hours.

Click Next> to display the Project Summary for your new project.

Click <Back to return to the Assign Data Collection Filters to Communication Links in Project page or Cancel to exit the New Project wizard without creating the new project.

Final Step: Project Summary

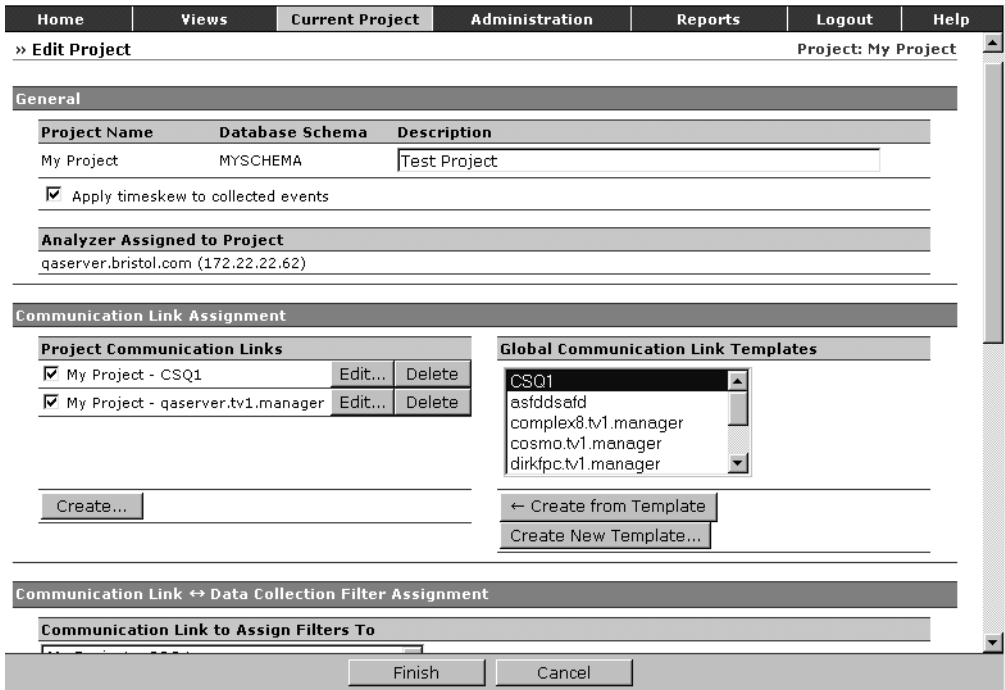The project summary lists the settings you have made for your new project.

Click Finish to confirm that the settings are correct and create the new project. The Projects page is displayed.

Click <Back to return to the Assign Data Collection Filters to Communication Links in Project page or Cancel to exit the Project wizard without creating the new project.

Edit a Project

Projects may only be edited by users with the necessary permissions. See Chapter 8, "Managing Users," for information about permissions.

To edit the current project, choose the Current Project > Project Status or Administration > Projects menu item and click Edit Project... to display the Edit Project page.

Home	Views	Current Project	Administration	Reports	Logout	Help

» **Edit Project**   Project: My Project

General

Project Name	Database Schema	Description
My Project	MYSCHEMA	Test Project

☑ Apply timeskew to collected events

Analyzer Assigned to Project
qaserver.bristol.com (172.22.22.62)

Communication Link Assignment

Project Communication Links		
☑ My Project - CSQ1	Edit...	Delete
☑ My Project - qaserver.tv1.manager	Edit...	Delete

Global Communication Link Templates

```
CSQ1
asfddsafd
complex8.tv1.manager
cosmo.tv1.manager
dirkfpc.tv1.manager
```

[Create...]

[← Create from Template]
[Create New Template...]

Communication Link ↔ Data Collection Filter Assignment

Communication Link to Assign Filters To

[Finish] [Cancel]

Make desired changes to the following project settings, then click Finish:

Project description

Enter a brief description for your project. This information appears on the TransactionVision home, Projects, and Project Status pages.

Time Skew

To apply the timeskew detected by TransactionVision on the communication link to events, check the Apply timeskew to collected events checkbox. This box is checked by default. Note that disabling the timeskew calculation affects all projects that use the database schema assigned to your project. You must restart the Analyzer associated with your project for this change to take effect.

Communication Link Assignment

The Analyzer used by a project must be assigned one or more communication links to allow a Sensor to communicate from a host on which an application is being monitored to the Analyzer.

- To assign a communication link to the Analyzer, select a communication link template in the list of global communication link templates and click <Create from Template. The communication link created for your project is a copy of the selected template; changes to the template do not affect the communication link used by your project.

- To view details about a global communication link template, choose the Administration > Communication Link Templates menu item to display the Communication Link Templates page.

- To add a new global communication link template, click Create New Template... to start the Communication Link wizard. See "Create a New Communication Link or Template" on page 64 for information about creating communication link templates.

- To create a communication link for the project that is **not** based on a global template, click Create... to start the Communication Link wizard. See "Create a New Communication Link or Template" on page 64 for information about creating communication links.

- To modify a communication link assigned to the Analyzer, click Edit.... to open the Edit Project Communication Link page. See "Edit a Communication Link or Template" on page 76 for information about editing communication links.

- To delete a communication link assigned to the Analyzer, click Delete.

- To disable a communication link without deleting it, clear the checkbox to the left of the communication link. To enable a disabled communication link, check the box.

Communication Link to Assign Filters To

You must assign one or more data collection filters to each communication link for the Analyzer assigned to your project. By default the Collect All filter, which collects all events, is assigned. Select the communication link to view/assign filters to from the drop-down list, then check filters to assign them or clear checks to unassign filters.

To create a new data collection filter definition, click Create New Filter... to display the Add Filter page. See "Create a New Data Collection Filter" on page 86 for information about creating new data collection filters.

Managing Communication Links

TransactionVision uses WebSphere MQ for communication between Sensors and the Analyzer. Before a Sensor can collect events and forward them to the Analyzer, you must create a communication link between the Sensor and the Analyzer. A communication link defines the message queues used to pass configuration and event messages between the Sensor and Analyzer.

Only users with required permissions may create or edit communication links. See Chapter 8, "Managing Users," for information about permissions.

Understanding Communication Links

TransactionVision communication links require two message queues for the exchange of messages between a Sensor and the Analyzer:

- A configuration queue that Sensors monitor for configuration messages from the Analyzer

- An event queue that the Analyzer monitors for event messages from Sensors

Configuration Queues

When TransactionVision begins recording events for a project, the Analyzer sends a configuration message to the configuration queues defined in the active communication links for the project. This configuration message defines the data collection filter conditions for the project, the name of the queue that event messages should be sent to, and the expiration time of the configuration message itself.

When an application makes a Sensored API call, it actually calls the Sensor library installed on the application host. The first time an application makes a Sensored API call, the Sensor checks to see whether a configuration queue exists on the queue manager identified by the MQI hConn parameter. If no configuration queue exists, the Sensor simply passes the call on to the standard technology library. If a configuration queue does exist, the Sensor reads all configuration messages on the queue, deleting any that have expired.

Event Queues

The Sensor evaluates the API call against the data collection filter conditions specified in the configuration message. If all filter conditions are met, the Sensor creates an event entry message about the API call. The Sensor then passes the API call on to the technology library. When the technology library returns from the call, the Sensor creates an event exit message, and then passes the return on to the calling application. Entry and exit events are grouped into a single message to be put on the specified event queue. There may be more than one configuration message, so event messages may be put on more than one event queue.

Sensors use the same configuration queue for all applications that make Sensored API calls on the same queue manager. However, data collection filter conditions such as host name or program name may limit the number of event messages sent to the Analyzer.

After the first Sensored API call made by a program instance, the Sensor continues to check the configuration queue periodically for new configuration messages.

While the Analyzer is recording events, it reads messages from the event queues into TransactionVision project schemas. It also sends new configuration messages to replace expiring configuration messages or to change the data collection filter conditions. When the Analyzer stops recording events, it sends a new configuration message to the configuration queue indicating that no more events should be recorded. The Analyzer may stop recording events for any of the following reasons:

- A user stops event recording from the TransactionVision web user interface.

- The end time specified in the data collection filter is reached.

Configuration and event queues may reside on the same queue manager or on different queue managers:

- Both the Sensor and the Analyzer communicate directly with the same queue manager.

- The Sensor uses one queue manager and the Analyzer uses a different one.

- The Sensor uses one queue manager, and the Analyzer sends configuration messages to one queue manager and monitors an event queue on a different queue manager.

Important! The configuration queue used by the WebSphere MQ Sensor must be on the same queue manager used by the application being monitored.

TransactionVision Queue Requirements

When creating queues for use with TransactionVision, you must specify attributes such as maximum message size and maximum queue depth. The settings you choose depend on factors such as the number of Analyzers using a queue manager, the number and size of messages used by monitored applications and the number of data collection filter conditions you specify.

In addition, TransactionVision event and configuration queues must have specific queue parameters and security permissions.

Message Length

The message length requirements for configuration queues are small. A default configuration message uses approximately 500 bytes. However, specifying data collection filter conditions increases the size of configuration messages. A minimum message length of 10,000 bytes is required.

The length of event messages can vary greatly, depending on the parameters passed to the API, the user data passed through the calls, and data collection filter conditions. For event queues, Bristol Technology recommends using the default maximum message length of 4194304 bytes; a minimum message length of 10,000 bytes is required. To restrict the message length based on your applications, consider the length of messages sent by your application and allow an additional 4000 bytes for event information.

Queue Depth

For configuration queues, the default queue depth should be sufficient. Analyzers send configuration messages to start data collection, to change data collection filter conditions, to end data collections, and to determine time skews across hosts. The number of configuration messages sent to a configuration queue depends on the number of Analyzers sending messages to the queue.

Event queue depth should be set to a value that matches the queue manager storage space based on the average event message size. It should be adequate to handle the peak volume. Bristol Technology recommends using the largest queue depth possible for the event queue and, in the case of remote communication link configurations, all queues in between. In most cases, Sensors put event messages on the event queues faster than Analyzers retrieve them. Insufficient queue depth may result in event queues filling up and Sensors slowing down waiting for the Analyzer to catch up. Increasing the event queue depth helps prevent this situation.

Event Queue Manager

It is best to have the actual event queue hosted on a queue manager other than the production queue manager, so that issues related to event queues (such as running out of disk storage due to event backlog) will not affect normal operations.

Event Queue Storage

Event queue storage should match the event queue depth based on the event message size. It should not exceed the storage capability of the queue manager host system. This is especially important when the event queue and TransactionVision dead letter queue are hosted by the production queue manager because running out of storage space may stop the queue manager completely.

Event Queue Message Persistency

The event queue message persistency property should match the event collecting policy.

Queue Parameters

- TransactionVision configuration and event queues must have the Default Share Option (DEFSOPT) set to SHARED.

- TransactionVision configuration and event queues must allow Shared Access (SHARE).

- TransactionVision configuration and event queues must allow Get (GET(ENABLED)) and Put (PUT(ENABLED)) operations.

- TransactionVision configuration queues must allow messages of at least 10,000 bytes in length (MAXMSGL).

- TransactionVision event queues must allow messages of at least 10,000 bytes in length (MAXMSGL).

- All channels (including client channels) must have MAXMSGL of at least 10,000 bytes.

Sensor Security Permissions

Userids of programs using the Sensor require the following:

- GET and BROWSE authority to the TransactionVision configuration queue.

- PUT authority to the TransactionVision event queue.

Analyzer Security Permissions

Users of the Analyzer require the following:

- PUT, GET, BROWSE, and INQ authority to all TransactionVision configuration and event queues.

- Passid permission to test communication links.

Note that Analyzer users can be given authority to only subsets of the configuration/event queues if you want to limit access to certain Sensors.

If a user does not have the required access permission on the configuration queue (TVISION.CONFIGURATION.QUEUE by default), the following message will be found in TransactionVision logs:

```
TransactionVision Sensor: cannot open event queue
TVISION.CONFIGURATION.QUEUE on queue manager merce.es1.manager:
Not authorized for access.
```

To set access permission of TVISION.CONFIGURATION.QUEUE for a user (tester in this example), run the following command:

```
setmqaut -m queuemanager -n TVISION.CONFIGURATION.QUEUE -t
queue -p tester. +get +browse +put +inq
```

It will return a message that the command completed successfully.

To display the access permission of TVISION.CONFIGURATION.QUEUE for a user (tester in this example), run the following command:

```
dspmqaut -m queuemanager -n TVISION.CONFIGURATION.QUEUE -t
queue -p tester
```

It will return output similar to the following example:

```
Entity tester has the following authorizations for object
TVISION.CONFIGURATION.QUEUE:
```

```
get
browse
put
inq
```

View Global Communication Link Templates

TransactionVision enables you to define global communication link templates to be used as the basis for project communication links. A global communication link template identifies the configuration and event queues and queue managers for the Sensor and the Analyzer.

When you select a global communication link template when creating a new project, project communication links based on the template are created. The project communication link is a copy of the global communication link template. Changes made to the template do not affect project commutation links and changes to project communication links do not affect the template.

Choose the Administration > Communication Link Templates menu item to display the Global Communication Link Templates page. Use this page to view, create, edit, test, and delete communication link templates.

The diagram provides a graphical representation of the selected communication link template.

New...

Click New... to start the Communication Link Wizard. This wizard guides you through the entire process of creating a new communication link template. See "Create a New Communication Link or Template" on page 64 for information about creating a new global communication link template.

Edit...

Click Edit... to display the Edit Communication Link Template page and make any desired changes. See "Edit a Communication Link or Template" on page 76 for information about modifying a communication link template.

Copy

Select an existing communication link and click Copy to create a copy of a communication link named Copy of *original_link_name*. Select the copy and click Edit... to change settings on the Edit Communication Link Template page.

Delete

To delete a communication link template, select it and click Delete. You will be prompted to confirm the deletion; click Yes to proceed.

Test

Perform the following steps to test a communication link:

1. Select the communication link to test.

2. Select the Analyzer to test the selected communication link with.

3. Click Test. The selected Analyzer sends a request message to the configuration queue requesting a confirmation on arrival (COA). The COA report message is sent to the event queue by the configuration queue that the communication link is configured to monitor. The end result is much like the behavior of typical communication between the Analyzer and Sensors. The communication link entry in the list of communication links is updated to show the Analyzer and test status.

If the test fails, the error message is displayed in the test status column. The following are the most common reasons for a communication link test to fail:

- Timeout waiting for a reply message
  This error is typically caused by channel problems with remote configurations. The request message may not make it to the final queue manager that is to report, or the report message may not make it to the queue manager the Communication Link Editor is connected to. If the link test fails for this reason, see Chapter 9, "Troubleshooting."

- WebSphere MQ errors
  WebSphere MQ problems within the communication link are reported with the WebSphere MQ error reason, which usually provides enough information to isolate the problem.

View Communication Links for a Project

Choose the Current Project > Communication Links menu item to display the Project Communication Links page. Use this page to view, create, edit, test, and delete communication links for the current project.

| Home | Views | **Current Project** | Administration | Reports | Logout |

» Project Communication Links   Project: My Pr

Communication Link Name	Queue Manager	Analyzer Tested On	Test Status	Time Sk
◉ My Project - qaserver.tv1.manager	qaserver.tv1.manager	172.22.22.62	Successful Test	00:00:00

Configuration of My Project - qaserver.tv1...

qaserver.tv1.manager

Configuration Queue
TVISION.CONFIGURATIO...

Sensor

Event Queue
TVISION.EVENT.QUEUE

Analyzer

| New... | Edit... | Copy | Delete | Analyzer: | qaserver.bristol.com (172.22.22.62) ▾ | Test |
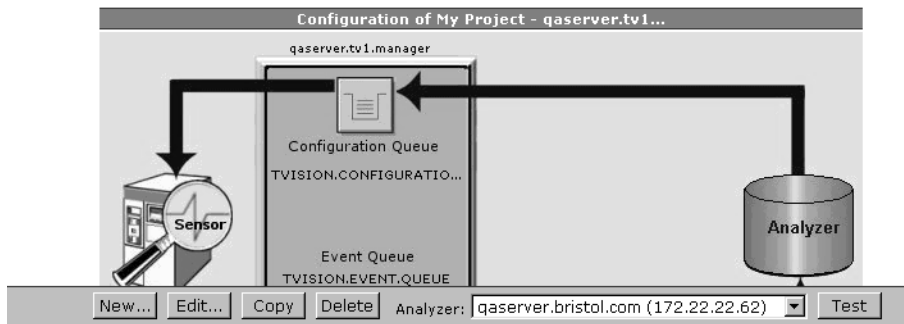
When you select a communication link, the diagram provides a graphical
representation of the selected communication link.

New...

Click New... to start the Communication Link Wizard. This wizard guides you through
the entire process of creating a new communication link. See "Create a New
Communication Link or Template" on page 64 for information about creating a new
global communication link template.

Edit...

Click Edit... to display the Edit Communication Link page and make any desired
changes. See "Edit a Communication Link or Template" on page 76 for information
about modifying a communication link template.

Copy

Select an existing communication link and click Copy to create a copy of a
communication link named Copy of *original_link_name*. Select the copy and click
Edit... to change settings on the Edit Communication Link page.

Delete

> To delete a communication link, select it and click Delete. You will be prompted to confirm the deletion; click Yes to proceed.

Test

> Perform the following steps to test a communication link:
>
> 1. Select the communication link to test.
>
> 2. Select the Analyzer to test the selected communication link with.
>
> 3. Click Test. The selected Analyzer sends a request message to the configuration queue requesting a confirmation on arrival (COA). The COA report message is sent to the event queue by the configuration queue that the communication link is configured to monitor. The end result is much like the behavior of typical communication between the Analyzer and Sensors. The communication link entry in the list of communication links is updated to show the Analyzer and test status.
>
> If the test fails, the error message is displayed in the test status column. The following are the most common reasons for a communication link test to fail:
>
> • Timeout waiting for a reply message
> This error is typically caused by channel problems with remote configurations. The request message may not make it to the final queue manager that is to report, or the report message may not make it to the queue manager the Communication Link Editor is connected to. If the link test fails for this reason, see Chapter 9, "Troubleshooting."
>
> • WebSphere MQ errors
> WebSphere MQ problems within the communication link are reported with the WebSphere MQ error reason, which usually provides enough information to isolate the problem.

Create a New Communication Link or Template

> The Communication Link Wizard guides you through the process of creating a project communication link or a communication link template.

Important! The Communication Link Wizard does not check whether a queue exists when creating communication links. However, you must create all queues before TransactionVision can use the communication link.

Step 1: Assign Name

Home	Views	Current Project	Administration	Reports	Logout	Help

» Global Communication Link Template Creation Wizard -- Step 1: Assign Name

Enter Communication Link Name

New Communication Link Name: `testserver.tv1.manager`

Instructions

This wizard takes you through the steps of creating a communication link to allow a TransactionVision Sensor to communicate with a TransactionVision Analyzer. You will be asked a series of questions and prompted to enter the names of queue managers and queues used in the communication between the Sensor and the Analyzer.

There are two communication paths that need to be defined in the communication link--one path for configuration messages from the Analyzer to the Sensor and another path for captured events from the Sensor to the Analyzer.

Once defined, these links are selected and used in the Analyzer to control communication during the monitoring of applications on hosts where the Sensor is active.

[Next >]   [Cancel]

Enter the name of the new communication link, then click Next> to define the Sensor connection.

Step 2: Sensor Connection

As you provide information on this page, the diagram is updated to provide a graphical representation of the Sensor connection for the new communication link.

1. Specify the name of the queue manager Sensors will connect to. This queue manager must be used by applications to be monitored.

2. Specify the name of the queue that Sensors should check for configuration messages. This queue must be a local queue on the specified queue manager. Check Use Default Configuration Queue to use the default configuration queue name TVISION.CONFIGURATION.QUEUE. To use a different queue name, uncheck Use Default Configuration Queue and enter the queue name.

3. Specify the name of the queue that Sensors should send event messages to. This queue should not be used by applications other than TransactionVision. If this queue is a local queue, check Use Local Queue Manager. If it is not a local queue, uncheck Use Local Queue Manager and specify either the name of the transmission queue that has the same name as the remote queue manager or a queue manager alias that resolves to the remote queue manager. This method of sending messages is described in the WebSphere MQ Application Programming Guide (SC33-0807-08).

4. Click Next> to specify whether the Analyzer should connect to the same queue manager that the Sensors connect to, or <Back to return to the communication link name page.

Step 3: Analyzer Connection Question

If the Analyzer should connect to the same queue manager that WebSphere MQ applications connect to, select Yes. Click Next> to specify the maximum message length (see Step 6).

If not, select No. Click Next> to specify whether to use one or two remote queue managers (see Step 4).

Step 4: Remote Queue Connection Question
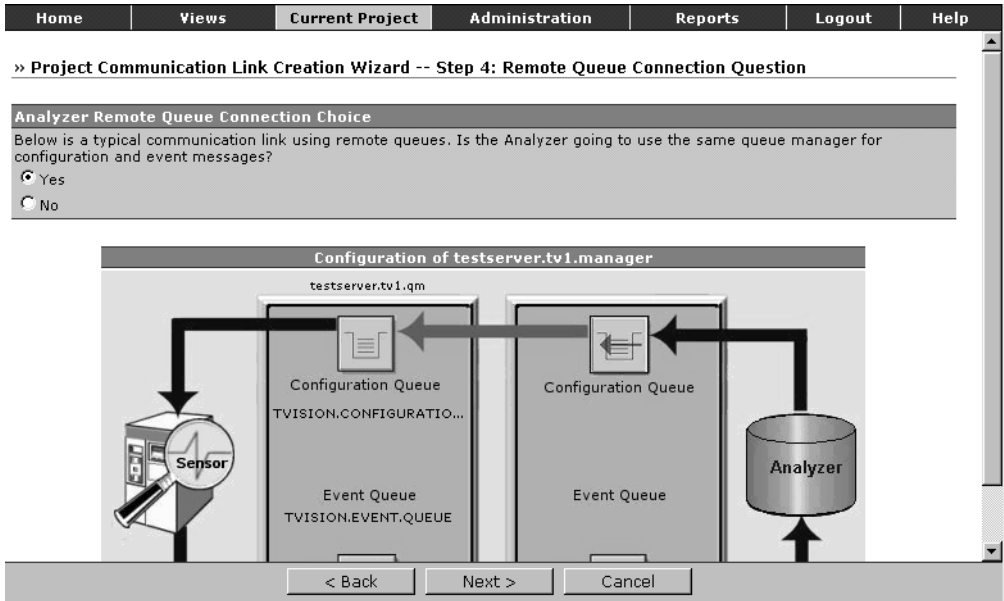
If the Analyzer should use the same queue manager for both configuration and event messages, select Yes.

If the Analyzer should use different queue managers for configuration and event messages,
select No.

Click Next> to specify remote queue connections or <Back to return to the Communication Link Analyzer Connection Question page.

Step 5: Remote Queue Connection

Same Queue Manager

If you answered Yes in Step 4, specify a single remote queue manager.

1. Specify the name of the queue manager the Analyzer will use for both configuration and event messages.

2. Specify the name of the queue that the Analyzer should send configuration messages to. This queue must be a remote queue for the configuration queue on the Sensor queue manager.

3. To use a transmission queue or queue manager alias, check the box next to User Transmission Queue or Queue Manager Alias. Enter the name of the transmission queue that has the same name as the remote queue manager or a queue manager alias that resolves to the remote queue manager.

4. Specify the name of the queue that the Analyzer should monitor for event messages. This queue must be a local queue and should not be used by applications other than TransactionVision.

5. Click Next> to specify the maximum message length or <Back to return to the Communication Link Remote Queue Connection Question page.

Separate Queue Managers

If you answered No in Step 4, specify separate remote queue managers.
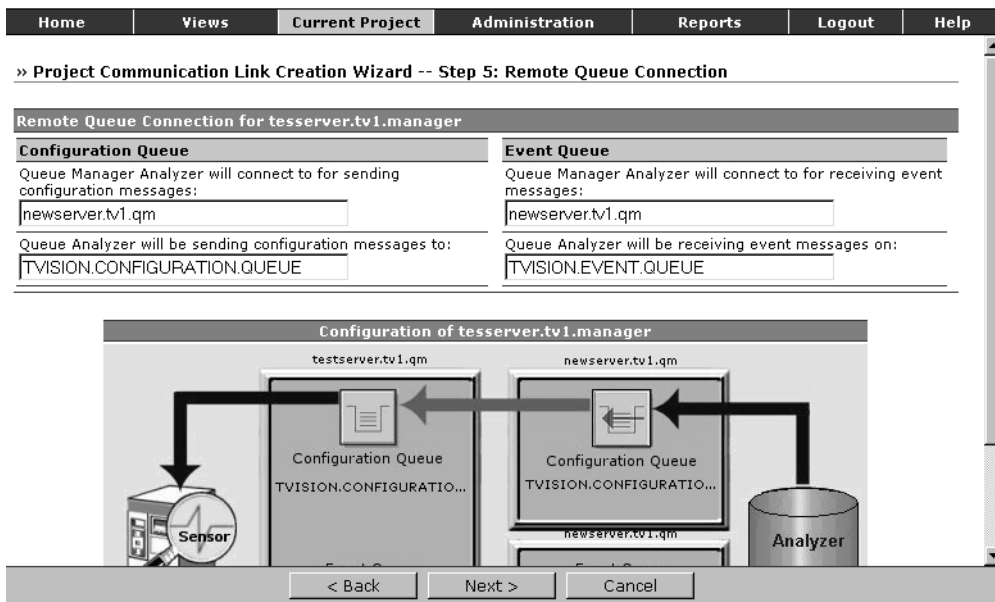
1. Specify the name of the queue manager the Analyzer will use for sending configuration messages to Sensors.

2. Specify the name of the queue that the Analyzer should send configuration messages to. This queue must be a remote queue for the configuration queue on the Sensor queue manager.

3. Specify the name of the queue manager the Analyzer will monitor for event messages from Sensors.

4. Specify the name of the queue on the event queue manager that the Analyzer should monitor for event messages. This queue must be a local queue and should not be used by applications other than TransactionVision.

5. Click Next> to specify the maximum message length or <Back to return to the Communication Link Remote Queue Connection Question page.

Step 6: Maximum Message Length

Specify the maximum event message length. This value can be no larger than the maximum message length of any of the event queues, channels, transmission queues, and queue managers used in this communication link.

Bristol Technology recommends using the default maximum message length of 4194304 bytes; a minimum message length of 10,000 bytes is required. To restrict the message length based on your applications, consider the length of messages sent by your application and allow an additional 4000 bytes for event information.

Click Next> to specify event delivery retry information for the communication link or <Back to return to the Remote Queue Connection page.

Step 7: Event Retry

The WebSphere MQ Sensor relies on the sensored application to handle connections to the event and configuration queue manager. However, other Sensors can be configured to try to reconnect if the event or configuration queue manager goes down and the Sensor loses the connection. Choose on of the following actions to take if the event or configuration connection is lost:

If a Sensor is unable to connect to the event queue or is unable to put events on the event queue, **events will be lost**. To avoid this situation, you can configure retry options for event connection and delivery.

Select one of the following actions for the Sensor to take if connection to the event queue fails:

Option	Description
Retry forever	Continues to retry at the specified interval as long as event collection is in progress. Note that this option may cause every application API call to slow down in the event of a connection failure, reducing application performance significantly. This option is recommended for an audit environment where all events must be logged.
Retry once per TransactionVision event	Retries once for each event. This option reduces the impact on application performance if a connection is lost, but TransactionVision events are lost if the retry attempt is not successful.
Retry across transaction events	Continues to retry at the specified interval as long as event collection is in progress. However, the reconnection time delay is not incurred for each API call, but at a given frequency.
Retry for a specified interval at every Transaction-Vision event	Continues to retry at the specified interval until the timeout is reached.

The default setting is to retry forever every 10 seconds.

To enable Sensors to retry if they fail to put an event on the event queue, check Retry Event Package Delivery on Failure. If you enable retry mode, set the following options to control retry attempts:

Option	Description
Retry Timeout	To continue to retry as long as event collection is in progress, choose Retry Forever. Otherwise, specify the number of seconds, minutes, or hours to continue to retry. The default is to retry forever.
Retry Interval	Specify the number of milliseconds or seconds that the Sensor should wait between failed retry attempts.

The default setting is not to retry event package delivery on failure. Set a value appropriate for your event collecting strategy. In a production environment, uncheck Retry Event Package Delivery on Failure for the safest setting.

To avoid any delays in the application when a failure occurs, setting Event Delivery Retry on Failure to not retry will override the Event Connection Retry on Failure option set on the same Communication Link.

Click Next> to specify miscellaneous information for the communication link or <Back to return to the Maximum Message Length page.

Final Step: Miscellaneous Information

Home	Views	Current Project	Administration	Reports	Logout	Help

» Project Communication Link Creation Wizard -- Final Step: Miscellaneous Information Inputs

Miscellaneous Information for tesserver.tv1.manager

Dead letter queue name for newserver.tv1.qm:	SYSTEM.DEAD.LETTER.QUEUE
Number of event collection threads (1-100):	1
Sensor trace logging:	○ On ◉ Off
Connection type:	○ Client ◉ Server

Configuration of tesserver.tv1.manager

testserver.tv1.qm   newserver.tv1.qm

Configuration Queue
TVISION.CONFIGURATIO...

Configuration Queue
TVISION.CONFIGURATIO...

Sensor

Analyzer

Event Queue
TVISION.EVENT.QUEUE

Event Queue
TVISION.EVENT.QUEUE

[< Back]   [Finish]   [Cancel]

1. Specify the name of the dead letter queue for the queue manager. The default is SYSTEM.DEAD.LETTER.QUEUE. If an event fails to be processed by the Analyzer, either because of an invalid event or an unexpected error or exception, the event is removed from the event queue and put on the dead letter queue defined in the communication link. Any fatal Java errors (such as OutOfMemoryError) will cause the event to be put on the dead letter queue and events will no longer be pulled off the event queue until the project is restarted. If the dead letter queue name is blank, the failed event will be discarded.

You can use the **GetPut** utility to transfer messages from the specified dead letter queue to enable the Analyzer to attempt to reprocess the failed events. See Appendix A, "Utilities Reference," for more information.

2. Specify the number of event collection threads, from 1 to 100. The default value is 1. This value should match the results from the DBMS insert test. For more information about performing this test, see *TransactionVision Installation and Configuration*.

Important! Multiple threads performing concurrent database operations along with a DB2 behavior called Next Row Locking may lead to database deadlocks. To avoid deadlocks when using multiple event-processing threads, set the DB2 variable DB2_RR_TO_RS to ON with the following command:

```
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.

3. Specify whether to enable Sensor trace logging. Sensor trace logging is off by default.

4. Specify whether to use a client or server connection type. The default connection type is server. If you select client, enter the following information:

Field	Description
Configuration Channel	Specify the channel to use when the Analyzer connects to the configuration queue manager.
Configuration Host	Specify the host running the configuration queue manager.
Configuration Port	Specify the listener port for the configuration queue manager.
Event Channel	Specify the channel to use when the Analyzer connects to the event queue manager.
Event Host	Specify the host running the event queue manager.

Field	Description
Event Port	Specify the listener port for the event queue manager.

5. Click Finish to create the communication link or <Back to return to the Communication Link Maximum Message Length page.

Edit a Communication Link or Template

Use the Edit Project Communication Link page to modify a project communication link or the Edit Communication Link Template page to modify a global communication link template.

Make desired changes to the following fields, then click Finish:

Field	Description
Communication link name	Enter the name of the communication link.
Sensor Connection	
Configuration Queue Manager	Specify the name of the queue manager Sensors will connect to. This queue manager must be used by applications to be monitored.
Configuration Queue	Specify the name of the queue that Sensors should check for configuration messages. This queue must be a local queue on the specified queue manager. The default configuration queue name is TVISION.CONFIGURATION.QUEUE.
Event Transmission Queue	Leave this field blank to indicate that the event queue is on the same queue manager as the configuration queue. If not, specify either the name of the transmission queue that has the same name as the remote queue manager or the queue manager alias that resolves to the remote queue manager. This method of sending messages is described in the WebSphere MQ Application Programming Guide (SC33-0807-08).
Event Queue	Specify the name of the queue that Sensors should send event messages to. This queue should not be used by applications other than TransactionVision.
Analyzer Connection	

Field	Description
Configuration Queue Manager	Specify the name of the queue manager the Analyzer will use for sending configuration messages to Sensors.
Configuration Transmission Queue	Specify either the name of the transmission queue that has the same name as the remote queue manager or the queue manager alias that resolves to the remote queue manager.
Configuration Queue	Specify the name of the queue that the Analyzer should send configuration messages to.
Event Queue Manager	Specify the name of the queue manager the Analyzer will monitor for event messages from Sensors.
Event Queue	Specify the name of the queue on the event queue manager that the Analyzer should monitor for event messages. This queue must be a local queue and should not be used by applications other than TransactionVision.
Event Connection Retry on Failure	
Retry forever	Continues to retry at the specified interval as long as event collection is in progress. Note that this option may cause every application API call to slow down in the event of a connection failure, reducing application performance significantly. This option is recommended for an audit environment where all events must be logged.
Retry once per TransactionVision event	Retries once for each event. This option reduces the impact on application performance if a connection is lost, but TransactionVision events are lost if the retry attempt is not successful.

Field	Description
Retry across transaction events	Continues to retry at the specified interval as long as event collection is in progress. However, the reconnection time delay is not incurred for each API call, but at a given frequency.
Retry for a specified interval at every TransactionVision event	Continues to retry at the specified interval until the timeout is reached.
Event Delivery Retry	
Retry Mode	To enable Sensors to retry if they fail to put an event on the event queue, check Retry Event Package Delivery on Failure. If you enable retry mode, set the Retry Timeout and Retry Interval to control retry attempts. To avoid any delays in the application when a failure occurs, setting Event Delivery Retry on Failure to not retry will override the Event Connection Retry on Failure option set on the same Communication Link.
Retry Timeout	To continue to retry as long as event collection is in progress, choose Retry Forever. Otherwise, specify the number of seconds, minutes, or hours to continue to retry. The default is to retry forever.
Retry Interval	Specify the number of milliseconds or seconds that the Sensor should wait between failed retry attempts.

Field	Description
Event Queue Maximum Message Length	Specify the maximum event message length. This value can be no larger than the maximum message length of any of the event queues, channels, transmission queues, and queue managers used in this communication link. Bristol Technology recommends using the default maximum message length of 4194304 bytes; a minimum message length of 10,000 bytes is required. To restrict the message length based on your applications, consider the length of messages sent by your application and allow and additional 4000 bytes for event information.
Dead Letter Queue Name	Specify the name of the dead letter queue for the queue manager where the Analyzer will place configuration messages. The default is TVISION.DEAD.LETTER.QUEUE. If left blank, the Analyzer will discard events that fail to be placed in the database. For more information about the dead letter queue, see "Final Step: Miscellaneous Information" on page 74.
Number of Event Collection Threads	Specify the number of event collection threads, from 1 to 100. The default value is 1. For more information about event collection threads, see "Final Step: Miscellaneous Information" on page 74.
Sensor Trace Logging	Specify whether to enable Sensor trace logging. Sensor trace logging is off by default.

Field	Description
Connection Type	Specify whether to use a client or server connection type. If you choose a client connection, you must also specify the following information:
	• The channel to use when the Analyzer connects to the configuration queue manager
	• The host running the configuration queue manager
	• The listener port for the configuration queue manager
	• The channel to use when the Analyzer connects to the event queue manager
	• The host running the event queue manager
	• The listener port for the event queue manager

Chapter 6

Managing Data Collection Filters

Collecting all possible event information about all API calls can cause the project database to grow quite large. It also adds to your WebSphere MQ overhead—depending on project settings, the Sensor may send a message for each event. However, you typically do not need to collect all information about every event that occurs in your system. For example, you may only need to collect events that occur on a specific host, events for a specific API, events with a specific MQI reason code, or events with a specific value in the user data buffer. You may require only a specific amount of user data for each event.

Data collection filters limit the number of events and the amount of data for each event that Sensors collect and forward to the Analyzer. The Analyzer communicates data collection filter criteria to Sensors via the configuration queue and queue manager defined in the communication links assigned to the Analyzer. The Sensor only collects events that match the filter criteria specified in the configuration message.

Only users with required permissions may create or edit data collection filters. See Chapter 8, "Managing Users," for information about permissions.

Planning Data Collection Filters

Data collection filters are unique for each project. You may create any number of filters for a project. You may assign a filter to multiple communication links, and you may assign multiple filters to a single communication link. When planning filters, keep the following in mind:

- An event only meets filter criteria if it matches **all** criteria specified for the filter. The AND operation is used **within** each filter.

- The Sensor collects events that meet the criteria for **any** assigned filter. The OR operation is used **between** filters.

- For each filter condition, you may specify whether to **include** or **exclude** events that meet the filter criteria.

- You may need to define more than one filter to meet your specific requirements.

Important! For the WebSphere MQ and CICS Sensors, a blank field is treated as a match when a data collection filter condition is specified. For example, if you create a filter that collects events with the user name J_SMITH, events with an empty user name are also collected by the WebSphere MQ and CICS Sensors.

Identifying Collection Needs

Before creating data collection filters, first identify exactly which events you need to collect. For example, you may only need to collect events for a particular Sensor type, host, or application. Or you may only need to collect events that use a specific WebSphere MQ object or application server. Or you may need to collect only events for a specific API, with a specific reason code, or with specific data in the user data buffer.

Once you determine which type of events you need to collect, you'll find it helpful to convert your criteria to an algebraic formula and generate a truth table to verify that the equation satisfies the requirement.

For example, suppose you want to collect all events from application X except MQCMIT events, and you also don't want to collect any event with the MQRC_NO_MSQ_AVAILABLE reason code. To state these requirements as an algebraic formula, use the following variables:

A = application X
B = APICODE MQCMIT
C = reason code MQRC_NO_MSQ_AVAILABLE

The resulting formula is as follows:

```
(A & !B & !C) | (!A & !C)
```

This formula shows two data collection filters. The first specifies the following conditions:

- Program = **include** events for application X

- API = **exclude** MQCMIT

- Reason code = **exclude** MQRC_NO_MSQ_AVAILABLE

The second filter specifies the following conditions:

- Program = **exclude** events for application X
- Reason code = **exclude** MQRC_NO_MSQ_AVAILABLE

The following truth table shows that this combination of data collection filters satisfies the filter requirements:
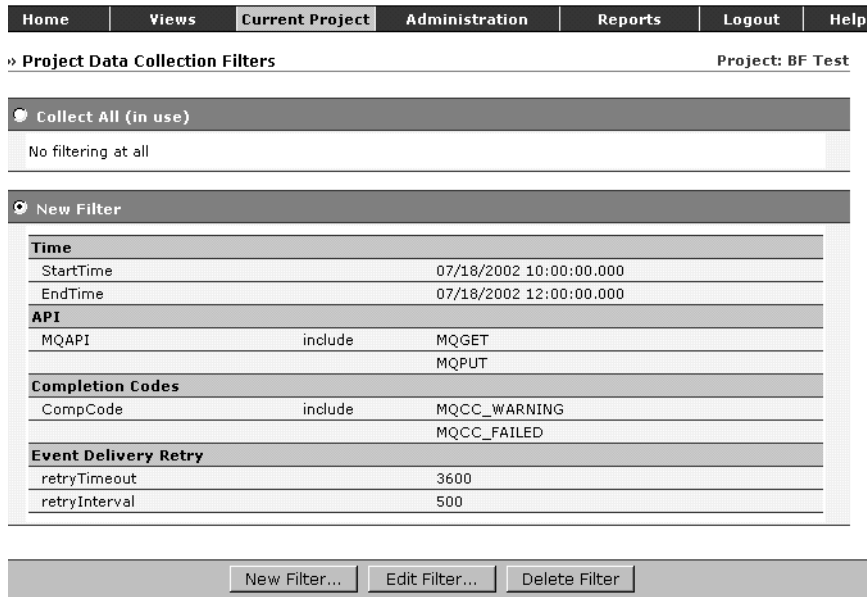
```
A B C    (A & !B & !C)  (!A & !C)   (A & !B & !C) | (!A & !C)
0 0 0    0              1           1
0 0 1    0              0           0
0 1 0    0              1           1
0 1 1    0              0           0
1 0 0    1              0           1
1 0 1    0              0           0
1 1 0    0              0           0
1 1 1    0              0           0
```

Data Collection Filters and Communication Links

Data collection filters assigned to communication links should have only the required MQ objects, applications, hosts, and APIs specified for collection. MQGET calls with NO_MSG_AVAIL reason code should generally be filtered out.

View Filters for the Current Project

The Project Data Collection Filters page displays a list of data collection filters available for the current project. To view this page, click Current Project > Data Collection Filter.

Home	Views	**Current Project**	Administration	Reports	Logout	Help

›› **Project Data Collection Filters**   Project: BF Test

⚪ Collect All (in use)
No filtering at all

⚫ New Filter		
Time		
StartTime		07/18/2002 10:00:00.000
EndTime		07/18/2002 12:00:00.000
API		
MQAPI	include	MQGET
		MQPUT
Completion Codes		
CompCode	include	MQCC_WARNING
		MQCC_FAILED
Event Delivery Retry		
retryTimeout		3600
retryInterval		500

[New Filter...] [Edit Filter...] [Delete Filter]

New Filter...

To create a new data collection filter, click New Filter... to display the Add Filter page. See "Create a New Data Collection Filter" on page 86 for information on creating a new filter.

Edit Filter...

To modify an existing filter, click the circle to the left of the filter name to select it. Click Edit Filter... to display the Edit Data Collection Filter page. See "Edit a Data Collection Filter" on page 89 for information about editing a data collection filter.

Delete Filter

To delete the selected filter, click Delete Filter.

Create a New Data Collection Filter

Perform the following steps to create a new data collection filter for a project:

1. Click New Filter... on the Project Data Collection filters page to display the Add Filter page.

| Home | Views | **Current Project** | Administration | Reports | Logout | Help |

» **Add Filter**

New Filter Options		
Filter Name:	New Filter	
Project:	BF Test	

[Next >] [Cancel]

2. Enter the name for the new data collection filter. This name used on the Project List, Project Status, and Edit Project pages.

3. Click Next> to display the Edit Data Collection Filter page and specify collection criteria.

The left side of the page lists all filter conditions that you may change. The right side shows the setting for the currently selected criteria.

4. Click the criteria category name on the left side for the condition that you wish to change from the default value and make desired changes on the right side of the page. See "Filter Conditions" on page 90 for instructions on setting criteria for each condition.

5. Click Set Criteria to save your settings for the current category. To clear changes for the current category, click Clear Criteria. To reset all categories to the default criteria, click Reset All Criteria.

6. When all query criteria are set, click Finish.

Edit a Data Collection Filter

Perform the following steps to edit a data collection filter for a project:

1. On the Project Data Collection Filters page, select the filter you wish to edit and click Edit Filter... to display the Edit Data Collection Filter page.

The left side of the page lists all filter conditions that you may change. The right side shows the setting for the currently selected criteria.

2.   Click the criteria category name on the left side for the condition that you wish to change and make desired changes on the right side of the page. See "Filter Conditions" on page 90 for instructions on setting criteria for each condition.

3.   Click Set Criteria to save your settings for the current category. To clear changes for the current category, click Clear Criteria. To reset all categories to the default criteria, click Reset All Criteria.

4.   When all query criteria are set, click Finish.

Filter Conditions

The left side of the Edit Data Collection Filter page lists all the criteria categories for filters. Only events that match the criteria for each category are displayed in the TransactionVision views.

Sensors:

Select the Sensor for which you wish to specify filter criteria from the drop-down list. You may specify criteria that apply to all Sensors, or you may specify criteria specific to the CICS, EJB, WebSphere MQ, JMS, or Servlet filters.

Check the box to the left of the Sensor to include events for the selected Sensor in your filter results, or clear the box to exclude events for the selected Sensor from your filter results.

Common Options:

The following criteria apply to all Sensors. However, if you specify a criteria for All Sensor, you may specify a different criteria for an individual Sensor. The setting for the individual Sensor overrides the setting for all Sensors. For example, you select All Hosts for All Sensor, but on the Host page for the JMS Sensor, you may select only the host Host1. JMS events originating on Host1 and other Sensor events originating on any host all match the filter criteria.

Host

1. Select whether to Include or Exclude selected hosts in the filter criteria.

2. Check the box next to each host you want to include specifically in the filter criteria, or check the box next to All Host to include all hosts in the filter criteria.

3. The host list is generated from events collected in the project database. To add a new host to the list, enter the host name and click Add. To remove a host name from the list, check the box next to the host name and click Delete.

Program/Servlet/EJB
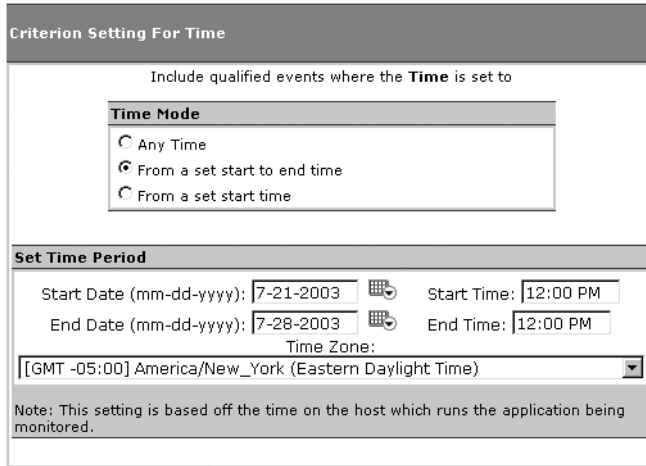
1. Select whether to Include or Exclude selected programs, servlets, or EJBs in the filter criteria.

2. Check the box next to each program, servlet, or EJB you want to include specifically in the filter criteria, or check the box next to All Program/Servlet/EJB to include all programs, servlets, and EJBs in the filter criteria.

3. The program list is generated from events collected in the project database. To add a new program to the list, enter the program name and click Add. To remove a program name from the list, check the box next to the program name and click Delete.

Time

Click the radio button next to the desired time mode.

Time Mode	Description
Any Time	Collects events with no time restrictions.
From a set start to end time	Collects events from a specific start time until a specific end time. Enter the start date and time, the end date and time, and select the time zone. To select the date from a calendar, click the calendar button and then click the desired date.
From set start time	Collects events from a specific start time until you explicitly stop recording. Specify the start date and time and select the time zone. To select the date from a calendar, click the calendar button and then click the desired date.

Important! Even if you specify a start time, you must initialize event recording before that start time. Initializing event recording causes the Analyzer to send a configuration message with filter information to all active communication links.

Data Range

1. Specify whether to collect only the API names called, the API names and call arguments, or the API names, call arguments, and user data buffer (All). Note that if you collect API names only, some TransactionVision functionality will not be available. You cannot view event details for events collected with this filter condition, the Analyzer cannot correlate MQPUT and MQGET calls, and you cannot specify API parameter values as query criteria in queries. However, restricting the amount of data collected for each event reduces the size of event messages.

2. If you specify a data range that includes user buffer data, select a range to specify how many bytes of the user data buffer to collect. Check the box next to All Data to collect the entire user data buffer.

3. To add a new range, enter the starting byte number in the Start Value field and the ending byte number in the End Value field and click Add. For example, use a start value of 0 and an end value of 100 to collect the first 100 bytes of the user data buffer. Specify a * in the End Value field to collect to the end of the user data buffer.

Important! Sensors collect data in all selected data range definitions, so deselect or remove any data ranges you do **not** want to collect. To remove an existing range, select it and click Delete.

Event Packaging

To enable Sensors to package multiple events together before sending them to the Analyzer, select Use Event Packaging. Event Packaging directly improves the performance of the Sensor. However, if the application terminates abnormally without doing an MQDISC, event packages buffered in the Sensor will **not** be sent to the Analyzer.

You may specify a number of events and/or maximum package size for event packaging. If you specify a value for both the number of events and the maximum package size, Sensors will combine message into packages based on which limit is reached first. If both are zero or blank, the default behavior of no event packaging is used.

Option	Description
Number of Events	Specify the number of events that will be stored in the Sensor before sending a package to the Analyzer. As soon as the Sensor reaches this number of events while packaging events together, it will send the package to the event queue.
	The number of event should be set to a value suitable to the environment, usually 10 events per message.
	Leave this field blank or specify a value of zero to use the maximum package size field exclusively.

Option	Description
Maximum Package Size	Specify the limit on the amount of memory a Sensor can set aside for storing events to be packaged. When the Sensor reaches or exceeds this limit while packaging events together, it will send the package to the event queue. Leave this field blank or specify a value of zero to use the number of events field exclusively. Note that if the Maximum Package Size is smaller than the event size, each event will be sent immediately. In this case, Event Packaging will effectively be turned off. The minimum event size is approximately 4000 bytes.

CICS Sensor Options:

The following options are available if you select the CICS Sensor:

API

This page allows you to filter CICS events by their API names as well as enable or disable CICS exits to filter a group of APIs.

1. Select whether to Include or Exclude events for selected APIs. To filter on selected exits, select Include and the API or API type check boxes for exits you wish to filter on. Selected APIs will be enabled; all other exits will be disabled. To disable specific exits, select Exclude and the API or API type check boxes for exits you wish to disable. All other exits will be enabled.

2. Check the box next to each API you want to enable or disable, or check the box next to All API to enable or disable all APIs. The APIs are grouped by API type. To select all APIs for a specific type, check the box next to the API type.

For example, if you select only the File Control exit to Include, all other exits are stopped and information messages are shown on the operator console. Similarly, if you Exclude File Control exits, all other exits are enabled.

CICS Transaction ID

1. Select whether to Include or Exclude events for selected CICS transaction IDs.

2. Check the box next to each transaction ID you want to include specifically in the filter criteria, or check the box next to All CICS Transaction ID to include all transaction IDs in the filter criteria.

3. The list of transaction IDs is generated from events collected in the project database. To add a transaction ID to the list, enter its name and click Add. To remove a transaction ID from the list, check the box next to it and click Delete.

CICS User ID

1. Select whether to Include or Exclude events for selected CICS user IDs.

2. Check the box next to each user ID you want to include specifically in the filter criteria, or check the box next to All CICS User ID to include all user IDs in the filter criteria.

3. The list of user IDs is generated from events collected in the project database. To add a user ID to the list, enter its name and click Add. To remove a user ID from the list, check the box next to it and click Delete.

File Name

1. Select whether to Include or Exclude events for selected filen names.

2. Check the box next to each file name you want to include specifically in the filter criteria, or check the box next to All File Name to include all file names in the filter criteria.

3. The list of file names is generated from events collected in the project database. To add a file name to the list, enter its name and click Add. To remove a file name from the list, check the box next to it and click Delete.

Response Code

1. Select whether to Include or Exclude events for selected response codes.

2. Check the box next to each response code you want to include specifically in the filter criteria, or check the box next to All Response Code to include all response codes in the filter criteria.

SYSID

1. Select whether to Include or Exclude events for selected CICS SYSIDs.

2. Check the box next to each SYSID you want to include specifically in the filter criteria, or check the box next to All CICS SYSID to include all SYSIDs in the filter criteria.

3. The list of SYSIDs is generated from events collected in the project database. To add a SYSID to the list, enter its name and click Add. To remove a SYSID from the list, check the box next to it and click Delete.

TD Queue Name

1. Select whether to Include or Exclude events for selected transient data (TD) queue names.

2. Check the box next to each TD queue name you want to include specifically in the filter criteria, or check the box next to All TD Queue Name to include all TD queue names in the filter criteria.

3. The list of TD queue names is generated from events collected in the project database. To add a TD queue name to the list, enter its name and click Add. To remove a TD queue name from the list, check the box next to it and click Delete.

TS Queue ID

1. Select whether to Include or Exclude events for selected temporary storage (TS) queue IDs.

2. Check the box next to each TS queue ID you want to include specifically in the filter criteria, or check the box next to All TS Queue ID to include all TS queue IDs in the filter criteria.

3. The list of TS queue IDs is generated from events collected in the project database. To add a TS queue ID to the list, enter its name and click Add. To remove a TS queue ID from the list, check the box next to it and click Delete.

Terminal ID

1. Select whether to Include or Exclude events for selected terminal IDs.

2. Check the box next to each terminal ID you want to include specifically in the filter criteria, or check the box next to All Terminal ID to include all terminal IDs in the filter criteria.

3. The list of terminal IDs is generated from events collected in the project database. To add a terminal ID to the list, enter its name and click Add. To remove a terminal ID from the list, check the box next to it and click Delete.

EJB Sensor Options:

The following options are available if you select the EJB Sensor:

Application Server

1. Select whether to Include or Exclude events for selected application servers.

2. Check the box next to each application server you want to include specifically in the filter criteria, or check the box next to All Application Server to include all application servers in the filter criteria.

3. The list of application servers is generated from events collected in the project database. To add an application server to the list, enter its name and click Add. To remove an application server from the list, check the box next to it and click Delete.

Application Name

1. Select whether to Include or Exclude events for selected application names.

2. Check the box next to each application name you want to include specifically in the filter criteria, or check the box next to All Application Name to include all application names in the filter criteria.

3. The list of application names is generated from events collected in the project database. To add an application name to the list, enter the name and click Add. To remove an application name from the list, check the box next to it and click Delete.

EJB Name

1. Select whether to Include or Exclude events for selected EJB names.

2. Check the box next to each EJB name you want to include specifically in the filter criteria, or check the box next to All EJB Name to include all application names in the filter criteria.

3. The list of EJB names is generated from events collected in the project database. To add a EJB name to the list, enter its name and click Add. To remove an EJB name from the list, check the box next to it and click Delete.

EJB Method

1. Select whether to Include or Exclude events for selected EJB methods.

2. Check the box next to each EJB method you want to include specifically in the filter criteria, or check the box next to All EJB Method to include all application names in the filter criteria.

3. The list of EJB methods is generated from events collected in the project database. To add an EJB method to the list, enter its name and click Add. To remove an EJB method from the list, check the box next to it and click Delete.

Exception Setting

Select whether to include both normal and exception events, normal events only, or exception events only.

JMS Sensor Options:

The following options are available if you select the JMS Sensor:

Web Application

1. Select whether to Include or Exclude events for selected web applications.

2. Check the box next to each web application you want to include specifically in the filter criteria, or check the box next to All Web Application to include all web applications in the filter criteria.

3. The list of web applications is generated from events collected in the project database. To add a web application to the list, enter its name and click Add. To remove a web application from the list, check the box next to it and click Delete.

Class

1. Select whether to Include or Exclude events for selected JMS classes.

2. Check the box next to each JMS class you want to include specifically in the filter criteria, or check the box next to All JMS Class to include all classes in the query criteria.

Method

1. Select whether to Include or Exclude events for selected JMS methods.

2. Check the box next to each JMS method you want to include specifically in the fil-
   ter criteria, or check the box next to All JMS Method to include all methods in the
   filter criteria.

3. Check Do not collect events when no message is available to avoid collecting
   events with no message.

Topic

1. Select whether to Include or Exclude events for selected topics.

2. Check the box next to each topic you want to include specifically in the filter crite-
   ria, or check the box next to All Topic to include all topics in the filter criteria.

3. The list of topics is generated from events collected in the project database. To add
   a topic to the list, enter its name and click Add. To remove a topic from the list,
   check the box next to it and click Delete.

JMS Queue

1. Select whether to include or exclude events for selected queues.

2. Check the box next to each queue you want to include specifically in the filter criteria, or check the box next to All Queue to include all queues in the filter criteria.

3. The list of queues is generated from events collected in the project database. To add a queue to the list, enter its name and click Add. To remove a queue from the list, check the box next to it and click Delete.

Connection Name

1. Select whether to include or exclude events for selected connection names.

2. Check the box next to each connection you want to include specifically in the filter criteria, or check the box next to All Connection to include all connections in the filter criteria.

3. The list of connections is generated from events collected in the project database. To add a connection to the list, enter its name and click Add. To remove a connection from the list, check the box next to it and click Delete.

Exception Setting

```
Criterion Setting For Exception Setting
collect both normal events and exception events ▼
```
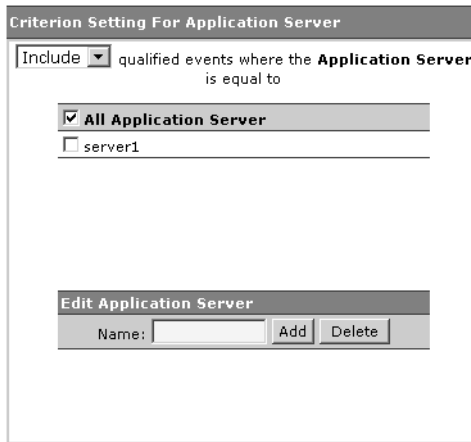
Select whether to include both normal and exception events, normal events only, or exception events only.

Servlet Sensor Options:

The following options are available if you select the servlet Sensor:

Application Server

```
Criterion Setting For Application Server
Include ▼ qualified events where the Application Server
            is equal to

☑ All Application Server
☐ server1



Edit Application Server
  Name: [          ]  Add   Delete
```

1. Select whether to Include or Exclude events for selected application servers.

2. Check the box next to each application server you want to include specifically in the filter criteria, or check the box next to All Application Server to include all application servers in the filter criteria.

3. The list of application servers is generated from events collected in the project database. To add an application server to the list, enter its name and click Add. To remove an application server from the list, check the box next to it and click Delete.

Web Application

```
Criterion Setting For Web Application

 Include ▼  qualified events where the Web Application is
                       equal to

   ☑ All Web Application
   ☐ Trade EJB




 Edit Web Application
     Name: [            ]  [ Add ] [ Delete ]

```

1. Select whether to Include or Exclude events for selected web applications.

2. Check the box next to each web application you want to include specifically in the filter criteria, or check the box next to All Web Application to include all web applications in the filter criteria.

3. The list of web applications is generated from events collected in the project database. To add a web application to the list, enter its name and click Add. To remove a web application from the list, check the box next to it and click Delete.

Servlet Method

```
Criterion Setting For Servlet Method

 Include ▼  qualified events where the Servlet Method is
                       equal to

   ☑ All Servlet Method         ▲
   ☐ destroy
   ☐ HTTP_DELETE
   ☐ HTTP_GET
   ☐ HTTP_HEAD
   ☐ HTTP_OPTIONS              ▼
```

1. Select whether to Include or Exclude events for selected servlet methods.

2. Check the box next to each servlet method you want to include specifically in the filter criteria, or check the box next to All Servlet Method to include all servlet methods in the filter criteria.

Client Host/IP

1. Select whether to Include or Exclude events for selected clients.

2. Check the box next to each client you want to include specifically in the filter criteria, or check the box next to All Client Host/IP to include all clients in the filter criteria.

3. The list of clients is generated from events collected in the project database. To add a client to the list, enter its name and click Add. To remove a client from the list, check the box next to it and click Delete.

Note that the Client Host/IP setting does not affect the init method.

URI

1. Select whether to Include or Exclude events for selected URIs.

2. Check the box next to each URI you want to include specifically in the filter criteria, or check the box next to All URI to include all URIs in the filter criteria.

3. The list of URIs is generated from events collected in the project database. To add a URI to the list, enter its name and click Add. To remove a URI from the list, check the box next to it and click Delete.

Status Code

1. Select whether to Include or Exclude events for selected status codes.

2. Check the box next to each status code you want to include specifically in the filter criteria, or check the box next to All Status Code to include all status codes in the query criteria.

Exception Setting

Select whether to include both normal and exception events, normal events only, or exception events only.

WebSphere MQ Sensor Options:

The following options are available if you select the WebSphere MQ Sensor:

WebSphere MQ API

1. Select whether to Include or Exclude selected WebSphere MQ APIs in the filter criteria.

2. Check the box next to each API you want to include specifically in the filter criteria, or check the box next to All API to include all WebSphere MQ APIs in the filter criteria.

3. Check Discard MQCMIT in empty transactions to avoid collecting empty transactions. Empty transactions occur in products like MQSI, which are in a loop doing MQGET calls on several queues. When where are no inputs, the MQGET calls fail with a NO_MSG_AVAILABLE and are typically filtered out by customers in TransactionVision. However, the MQGET is followed by an MQCMIT. This option provides an easy way to filter out these MQCMIT events.

4. Check Do not send MQDISC exit event to report only the entry information of each MQDISC call. This improves performance and reduces overhead for the Sensor. Otherwise, the Sensor must reconnect and reopen the event queue in order to send MQDISC exit information, and this information is typically not necessary.

Completion Code

1. Select whether to include or exclude events for selected WebSphere MQ API completion codes.

2. Check the box next to each API completion code you want to include specifically in the filter criteria, or check the box next to All Completion Codes to include all WebSphere MQ completion codes in the filter criteria.

Reason Code

Criterion Setting For Reason Code

Include ▼ qualified events where the **Reason Code** is equal to

☑ **All Reason Code**
☐ MQRC_ALIAS_BASE_Q_TYPE_ERROR
☐ MQRC_ALREADY_CONNECTED
☐ MQRC_BACKED_OUT
☐ MQRC_BUFFER_ERROR
☐ MQRC_BUFFER_LENGTH_ERROR

1. Select whether to include or exclude events for selected WebSphere MQ API reason codes.

2. Check the box next to each API reason code you want to include specifically in the filter criteria, or check the box next to All Reason Codes to include all reason codes in the filter criteria.

Tip! MQGET calls with a NO_MSG_AVAIL reason code should typically be filtered out. When monitoring WMQI brokers, make sure to filter out API calls to WMQI system queues MQGET with a NO_MSG_AVAIL reason code.

Object

1. Select whether to Include or Exclude selected objects in the filter criteria.

2. Check the box next to each queue manager or object you want to include specifically in the filter criteria, or check the box next to All Objects to include all objects in the filter criteria. Selecting objects is useful if the same object name is present on multiple queue managers (for example, as in clusters).

3. The object list is generated from events collected in the project database. To add a new queue manager or object to the list, enter the name and click Add. To remove a queue manager or object name from the list, check the box next to it and click Delete.

   You may use wildcards at the start and/or end of queue manager or object names to specify a group of queue managers or objects. To use wildcards, add a new queue manager or object using wildcards in the name. For example, to collect events on all queue managers called ES1.*, add a new queue manager called ES1.* and check the box next to it.

ReplyTo Object

1. Select whether to Include or Exclude selected Reply To queue managers and objects in the filter criteria.

2. Check the box next to each queue manager or queue you want to include specifically in the filter criteria, or check the box next to All Reply To Object to include all objects in the filter criteria. Selecting objects is useful if the same object name is present on multiple queue managers (for example, as in clusters).

3. The object list is generated from events collected in the project database. To add a new queue manager or object to the list, enter the name and click Add. To remove a queue manager or object name from the list, check the box next to it and click Delete.

   You may use wildcards at the start and/or end of queue manager or object names to specify a group of queue managers or objects. To use wildcards, add a new queue manager or object using wildcards in the name. For example, to collect events on all queue managers called ES1.*, add a new queue manager called ES1.* and check the box next to it.

MQI Broker

1. Select whether to Include or Exclude selected WebSphere MQ Integrator (MQI) brokers in the filter criteria.

2. Check the box next to each broker you want to include specifically in the filter criteria, or check the box next to All MQI Brokers to include all brokers in the filter criteria.

3. The broker list is generated from events collected in the project database. To add a new broker to the list, enter the name and click Add. To remove a broker name from the list, check the box next to it and click Delete.

For more information about integrating TransactionVision with WebSphere MQ Integrator, see the *TransactionVision User's Guide*.

MQI Message Flow

1. Select whether to Include or Exclude selected WebSphere MQ Integrator (MQI) message flows in the filter criteria.

2. Check the box next to each message flow you want to include specifically in the filter criteria, or check the box next to All MQI Message Flows to include all programs in the filter criteria.

3. The message flow list is generated from events collected in the project database. To add a new message flow to the list, enter the name and click Add. To remove a message flow name from the list, check the box next to it and click Delete.

For more information about integrating TransactionVision with WebSphere MQ Integrator, see the *TransactionVision User's Guide*.
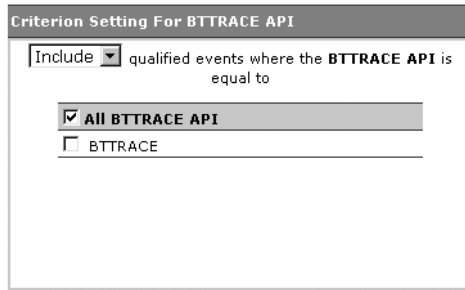
MQ IMS Bridge API
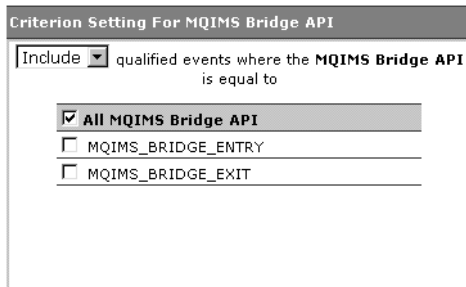
1. Select whether to include or exclude events for selected MQSeries-IMS bridge APIs.

2. Check the box next to the APIs you want to include specifically in the filter criteria, or check the box next to All API to include all MQSeries-IMS bridge APIs in the filter criteria.

For more information about the MQSeries-IMS Bridge Sensor, see the *TransactionVision Installation and Configuration Guide*.

BTTRACE API
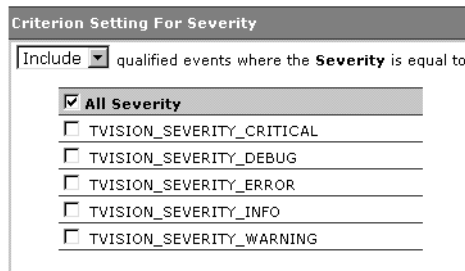
1. Select whether to include or exclude events for selected BTTRACE APIs. Applications may use the BTTRACE function provided by the Sensor library to send user-defined trace messages to the Analyzer.

2. Check the box next to each API you want to include specifically in the query criteria, or check the box next to All API to include all BTTRACE APIs in the query criteria.

For more information about the **BTTRACE** function, see Chapter 10, "Extending TransactionVision Sensors."

BTTRACE Severity

1. Select whether to include or exclude events for selected BTTRACE severity levels. Applications may use the BTTRACE function provided by the Sensor library to send user-defined trace messages to the TransactionVision analysis views.

2. Check the box next to each severity you want to include specifically in the query criteria, or check the box next to All Severity to include all BTTRACE severity levels in the query criteria.
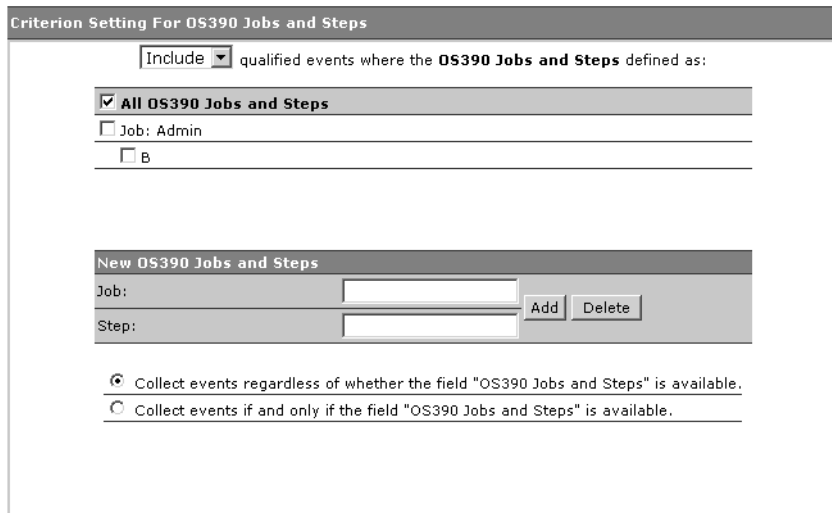
For more information about the **BTTRACE** function, see Chapter 10, "Extending TransactionVision Sensors."

User

1. Select whether to include or exclude events with selected user names.

2. Check the box next to each user name you want to include specifically in the filter criteria, or check the box next to Select All User Name to include all user names in the filter criteria.

3. The user name list is generated from events collected in the project database. To add a new user name to the list, enter the user name and click Add. To remove a user name from the list, check the box next to the user name and click Delete.

OS/390 Jobs and Steps
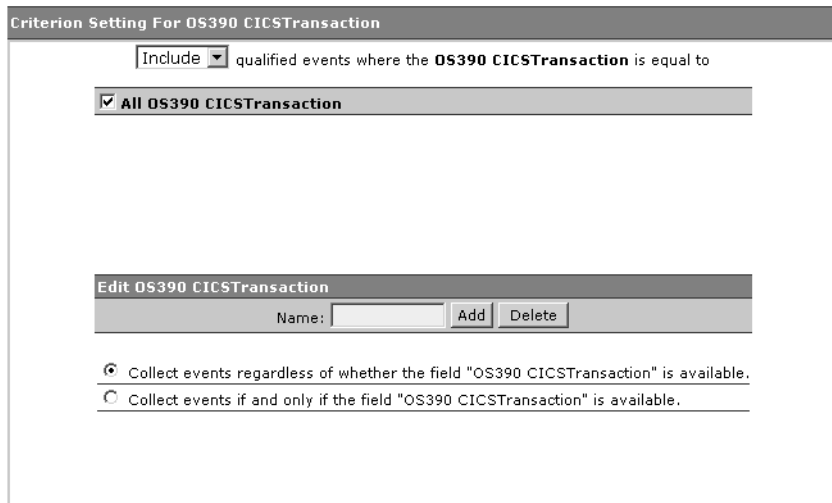
1. Select whether to Include or Exclude selected OS/390 job steps in the filter criteria.

2. Check the box next to each job name or step you want to include specifically in the filter criteria, or check the box next to All Jobs and Steps to include all job steps in the filter criteria.

3. The job name and job step list is generated from events already collected in the project database. If no events have yet been collected, enter the job name and click Add or select a job name, enter a job step and click Add to add a step to an existing job. To remove a job name or step from the list, select it and click Delete.

4. Only events generated on the OS/390 platform have job names and job steps. To collect events that do not have a job step (non-OS/390 events), choose to collect events regardless of whether the field exists.

OS/390 CICS SYSID

1. Select whether to include or exclude events for selected OS/390 CICS SYSIDs.

2. Check the box next to each CICS SYSID you want to include specifically in the filter criteria, or check the box next to All OS390 CICS SYSID to include all CICS SYSIDs in the filter criteria.

3. The list of CICS SYSIDs is generated from events collected in the project database. If no events have yet been collected, enter the CICS SYSID and click Add. To remove a SYSID name from the list, select it and click Delete.

4. Only events generated on the OS/390 CICS platform have a CICS SYSID. To collect events that do not have a CICS SYSID (non-OS/390 CICS events), choose to collect events regardless of whether the field exists.

OS/390 CICS Transaction

1. Select whether to include or exclude events for selected OS/390 CICS transactions.

2. Check the box next to each CICS transaction you want to include specifically in the filter criteria, or check the box next to All Transactions to include all CICS transactions in the filter criteria.

3. The list of CICS transactions is generated from events collected in the project database. If no events have yet been collected, enter the CICS transaction and click Add. To remove a transaction name from the list, select it and click Delete.

4. Only events generated on the OS/390 CICS platform have a CICS transaction. To collect events that do not have a CICS transaction (non-OS/390 CICS events), choose to collect events regardless of whether the field exists.

OS/390 IMS Region Type

1. Select whether to include or exclude events for selected OS/390 IMS region types.

2. Check the box next to each IMS region type you want to include specifically in the filter criteria, or check the box next to All Region Type to include all IMS region types in the filter criteria.

3. The list of IMS region types is generated from events collected in the project database. If no events have yet been collected, enter the IMS region and click Add. To remove a region type name from the list, select it and click Delete.

4. Only events generated on the OS/390 IMS platform have an IMS region type. To collect events that do not have an IMS region type (non-OS/390 IMS events), choose to collect events regardless of whether the field exists.

OS/390 IMS Region Identifier

1. Select whether to include or exclude events for selected OS/390 IMS region identifiers.

2. Check the box next to each IMS region identifier you want to include specifically in the filter criteria, or check the box next to All Region Identifier to include all IMS region identifiers in the filter criteria.

3. The list of IMS region identifiers is generated from events collected in the project database. If no events have yet been collected, enter the IMS region identifier and click Add. To remove a region identifier name from the list, select it and click Delete.

4. Only events generated on the OS/390 IMS platform have an IMS region identifier. To collect events that do not have an IMS region identifier (non-OS/390 IMS events), choose to collect events regardless of whether the field exists.

OS/390 IMS Identifier
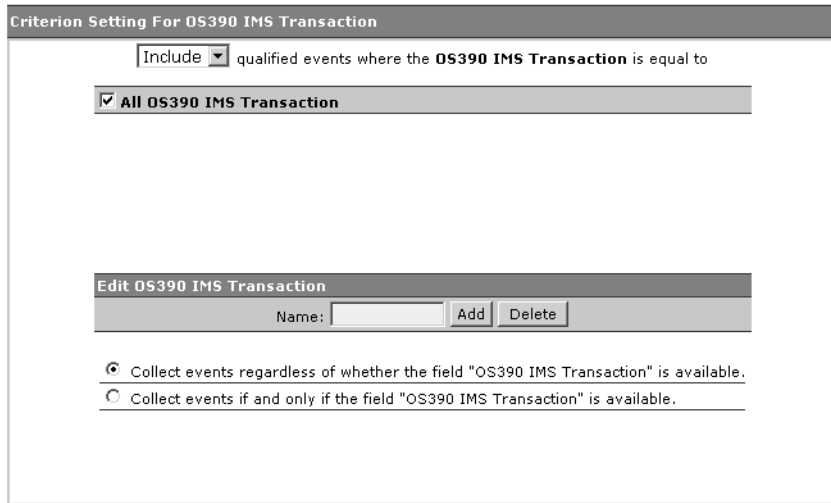
1. Select whether to include or exclude events for selected OS/390 IMS identifiers.

2. Check the box next to each IMS identifier you want to include specifically in the filter criteria, or check the box next to All Identifier to include all IMS identifiers in the filter criteria.

3. The list of IMS identifiers is generated from events collected in the project database. If no events have yet been collected, enter the IMS identifier and click Add. To remove an identifier name from the list, select it and click Delete.

4. Only events generated on the OS/390 IMS platform have an IMS identifier. To collect events that do not have an IMS identifier (non-OS/390 IMS events), choose to collect events regardless of whether the field exists.

OS/390 IMS Transaction

1. Select whether to include or exclude events for selected OS/390 IMS transactions.

2. Check the box next to each IMS transaction you want to include specifically in the filter criteria, or check the box next to All Transaction to include all IMS transactions in the filter criteria.

3. The list of IMS transactions is generated from events collected in the project database. If no events have yet been collected, enter the IMS transaction and click Add. To remove a transaction name from the list, select it and click Delete.

4. Only events generated on the OS/390 IMS platform have an IMS transaction. To collect events that do not have an IMS transaction (non-OS/390 IMS events), choose to collect events regardless of whether the field exists.

OS/390 IMS PSB

1. Select whether to include or exclude events for selected OS/390 IMS PSBs.

2. Check the box next to each IMS PSB you want to include specifically in the filter criteria, or check the box next to All PSB to include all IMS PSBs in the filter criteria.

3. The list of IMS PSBs is generated from events collected in the project database. If no events have yet been collected, enter the IMS PSB and click Add. To remove a PSB name from the list, select it and click Delete.

4. Only events generated on the OS/390 IMS platform have an IMS PSB. To collect events that do not have an IMS PSB (non-OS/390 IMS events), choose to collect events regardless of whether the field exists.

OS/400 Job

1. Select whether to include or exclude events for selected OS/400 job names.

2. Check the box next to each job name you want to include specifically in the filter criteria, or check the box next to All Job Names to include all job names in the filter criteria.

3. The list of job names is generated from events collected in the project database. If no events have yet been collected, enter the job name and click Add. To remove a job name from the list, select it and click Delete.

4. Only events generated on the OS/400 platform have an OS/400 job name. To collect events that do not have an OS/400 job name (non-OS/400 events), choose to collect events regardless of whether the field exists.

Managing Scheduled Jobs

A job is a task that runs at a specified frequency. A job typically gathers statistics of recently arrived events and stores calculated results in a way that is easily accessible by a report. By using a job, the reports themselves do not have to perform complex, time-consuming queries to present report data. Instead, they use already calculated data that is periodically updated by a job running in the background.

A job is a bean that implements a particular task. See the *TransactionVision Programmer's Guide* for information about creating job beans and, if you are using WebLogic, adding job beans to the TransactionVision application.

Use the TransactionVision web user interface to define and manage jobs. A job instance is defined for a particular project; that instance can only be run on that project. However, multiple instances of the job bean may be run concurrently, each configured for a different project.

Only users with required permissions may manage jobs. See Chapter 8, "Managing Users," for information about permissions.

Available Jobs

Transaction provides two jobs:

- deleteEvents

- Transaction Statistics

deleteEvents
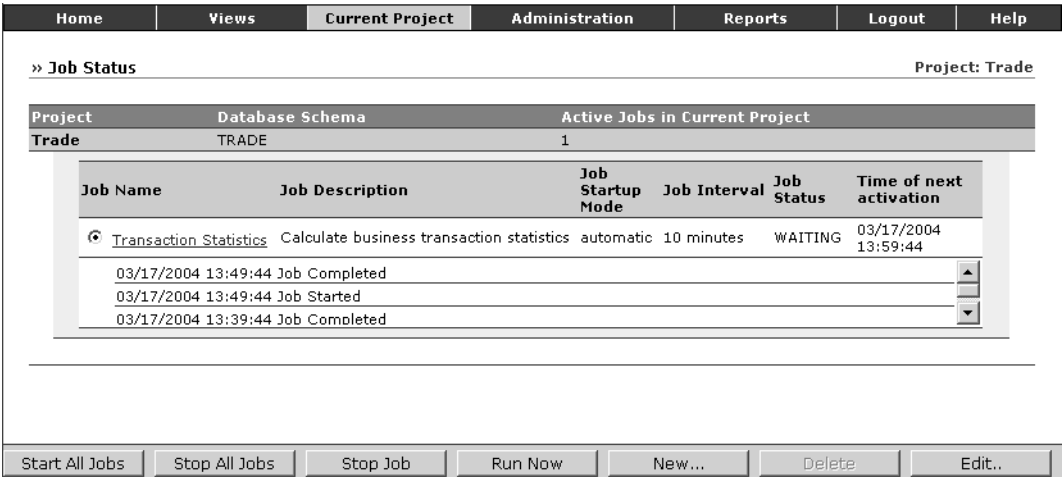
> The deleteEvents job deletes all events more than 48 hours old from your project database. This job is optional, and by default is turned off when you create a project. The deleteEvents job uses the **deleteEvents** utility; for detailed information about this utility, see Appendix A, "Utilities Reference."

Transaction Statistics

> The Transaction Statistics job calculates business transaction statistics.This job is required if you wish to collect transaction statistics; these statistics are used by a number of different reports. If this job is not running, those reports will not be able to show any data.

View Jobs for the Current Project

> Choose the Current Project > Job Status menu item to view the Job Status page. Use this page to view information about jobs for the current project, start or stop jobs, run waiting jobs, create new jobs, and delete jobs.

> This page shows the project name, assigned database schema, and number of active jobs. For each job, it provides the following information:
>
> • Job name. Click the job name link to edit an existing job configuration. For more information about editing a job configuration, see "Create/Edit a Job" on page 133.
>
> • Job description (optional)

- Job startup mode. The following table shows valid values:

Startup Mode	Description
Automatic	The job is automatically initialized and started when the web server starts or when a user clicks Start All.
Manual	The job is only started when the job is selected and started with the Start Job button.
Disabled	The job cannot be run until its startup mode is changed to Automatic or Manual. Disabling a job is useful if you have an automatically starting job that you want to temporarily turn off so it doesn't start when the system comes up or when starting all jobs.

- Job interval. Specifies the time period interval at which the job is configured to run.
- Job status. The following table shows valid values:

Status	Description
Running	The job is in the process of executing the task it is programmed to do.
Waiting	The job is active and is waiting for its scheduled run time to arrive.
Quiescing	The job is in the process of shutting down, performing any specific cleanup or shutdown processes applicable to the job. A job initially transfers to this state when it is stopped, then automatically move from this state to Stopped when the shutdown or cleanup process is complete. For example, the DeleteEvents job finishes making database updates for its current progress in this state. When in Quiescing mode, use the Force Stop button to end a job immediately without completing pending tasks.

Status	Description
Stopped	The job is not active, is not loaded into memory, and will not perform any tasks until started with the Start Job or Start All Jobs button.

- Time of next activation. Specifies the time a waiting or running job will be run next.

For the selected job, the Job Status page also displays history information showing when the job has been initialized, run, and completed.

Start All Jobs

To start all automatic jobs, click Start All Jobs. Starting a job initializes and activates it. If it's next scheduled time has already passed, it will run immediately; otherwise, it will be placed in a waiting state until its scheduled time arrives.

Stop All Jobs

To shutdown all currently active jobs, click Stop All Jobs. When a job is stopped, a cancel command is first issued, if the job is currently running, followed by commands to stop and shut it down.

Start/Stop Job

To start or stop a single job, click the radio button to the left of the job name to select it, then click Start Job to start a stopped job or Stop Job to stop a running or waiting job.

Force Stop

To force a job in the Quiescing state to stop immediately without completing pending cleanup or shutdown tasks, click the radio button to the left of the job name to select it, then click Force Stop. This button is only enabled if the selected job is in Quiescing mode.

Run Now/Cancel Run

To force a waiting job to run immediately, rather than continuing to wait for the scheduled time, click the radio button to the left of the job name to select it and click Run Now. To cancel the current run of a running job without stopping the job, select it and click Cancel Run. The job will change to a waiting state until the next scheduled run time.

New...

To create a new job configuration, click New... to open the Create Job page. For more information, see "Create/Edit a Job" on page 133.

Delete

To delete a job, click the radio button to the left of the job name to select it and click Delete. You may only delete a job if its status is stopped.

Edit...

To modify a job configuration, click Edit Job... to open the Edit Job page. For more information, see "Create/Edit a Job" on page 133.

View Jobs for All Projects

To view job information for any project, rather than for the current project only, choose the Administration > Jobs menu item. The Job Status page opens. This page is similar to the Job Status page for the current project, except it has a list of projects, with the current project selected. To view job status information for a different project, click the radio button to the left of the project name to select it. For more information about job status, see "View Jobs for the Current Project" on page 130.

Create/Edit a Job

To create a new job or edit an existing job, you must first make the job class accessible to the TransactionVision application in your application server, then you must add it in the TransactionVision user interface.

Make a Job Accessible in an Application Server

Before creating a new job, add the job class under in you application server to make the class file accessible to the TransactionVision application.

WebSphere

For WebSphere, do **either** of the following:

- Set the CLASSPATH for your server through the WebSphere Administration Console to include your class file by choosing Process definition > Java virtual machine.

- Place the job class file in the `WAS_HOME/installedApps/.../TransactionVision.ear/tvision/war/WEB-INF/classes` directory in an appropriate subdirectory for its class name.

WebLogic

To add a custom job class to WebLogic, which does not have an `installedApps` directory, perform the following steps to repackage and redeploy the TransactionVision application to include the relevant classes and/or jar files:

1. Make a backup copy of the `tvision.ear` file in `TVISION_HOME/ui`.

2. Copy the `tvision.ear` file to a temporary location and extract it using the following command:

    ```
    jar -xf tvision.ear
    ```

    A `tvision.war` file will be among the files extracted.

3. Move the `tvision.war` file to a temporary location and extract it as well using the following command:

    ```
    jar -xf tvision.war
    ```

    This command will extract all the TransactionVision related files. Within the files extracted, you will find two directories: `WEB-INF/lib` and `WEB-INF/classes`.

4. Copy any required jar files required to the `WEB-INF/lib` directory and any required class files to the `WEB-INF/classes` directory.

    Note: If class files are part of a package, they must be placed in that corresponding directory. For example, a java class called `com.mycompany.TestClass` would need to be placed in the `com/mycompany` directory within `WEB-INF/classes`.

5. Repack the `tvision.war` file. Remove the old `tvision.war` file from the directory and the recreate it, packing all the files in the directory by using the following command:

    ```
    jar -cf tvision.war
    ```

6. Copy this new `tvision.war` file to the directory where you extracted the `tvision.ear` file.

7. Repack the `tvision.ear` file in this directory with the following command:

    ```
    jar -cf tvision.ear
    ```

8. Copy this new ear file to `TVISION_HOME/ui`.

9. Run **TVisionSetup** to redeploy TransactionVision. The new jobs classes should now be available to the web application.

Add a Job to the TransactionVision User Interface

Once you have made your job accessible to TransactionVision in your application server, perform the following steps to add it to the TransactionVision user interface:

1. To create a new job, click *New...* on the Job Status page. To edit an existing job, select the job name on the Job Status page and click *Edit...*. TransactionVision displays the Create/Edit Job page.

2. Enter the job name and optional description as you want them to appear on the Job Status page.

3. Select one of the following startup modes:

Startup Mode	Description
Automatic	The job is automatically initialized and started when the web server starts or when a user clicks Start All.
Manual	The job is only started when the job is selected and started with the Start Job or Run Now button.
Disabled	The job cannot be run until its startup mode is changed to Automatic or Manual. Disabling a job is useful if you have an automatically starting job that you want to temporarily turn off so it doesn't start when the system comes up or when starting all jobs.

4. Enter the name of the java class that is to be instantiated for this job in the Class Name field. This class must implement the IJob interface (see the *TransactionVision Programmer's Guide* for details).

5. Enter any optional startup parameters. They are passed to the job bean's initialization method when it starts. For the deleteEvents job, keep the following in mind:

   • By default, the value for the -commit option for the deleteEvents job is 0, meaning that the commit is done after the entire deletion. If you are deleting a large number of events, explicitly specify the parameter -commit *n* to commit after *n* event deletions. Otherwise, the database transaction log may overflow or the table may be locked for a long period of time.

   • By default, the deleteEvents job and script do not allow deleting events in the last 48 hours. To override the default of 48 hours, use the -minage number-of-hours option. For example, -minage 10 prevents the job from deleting events from the last 10 hours.

6. Enter the frequency of when the job should be repeated. The interval may be in minutes, hours, days, or months.

7. Enter the next scheduled job time, if desired. If you do not enter a next scheduled time, the next scheduled time is calculated based on the information in the Repeat Every field.

8. If you want to propagate this job to any existing projects, select the projects in which to create this job.

9. Click Finish to return to the Job Status page.

TransactionVision uses a Lightweight Directory Access Protocol (LDAP) server to authenticate users and store permissions for what TransactionVision operations a user is or is not allowed to perform. If your organization uses an existing LDAP infrastructure, you can easily integrate TransactionVision's user management.

If you do not have an LDAP server, or if you wish to keep TransactionVision separate from your existing server, download the IBM Directory Server 5.1 server from the following location: http://www-3.ibm.com/software/tivoli/products/directory-server/.

Important! If you are using the LDAP server embedded with BEA WebLogic, see "Using the WebLogic LDAP Server" on page 141 for additional information.

When you run the **TVisionSetupInfo** and **TVisionSetup** utilities during installation, the information you provide is used to set entries in your TransactionVision LDAP configuration file. In addition, the TVAuthorityGroup attribute, TVAuthorization class, and TVPerson class are added to your LDAP server's object class and attribute schema. Finally, a hierarchy named ou=TVision containing predefined security groups and a user entry for allowing an administrator to log in are created at the specified location.

User Attributes

To assign TransactionVision permissions to users, assign a user or group of users to a TVAuthorization object. This object contains all the attributes indicating what the user is allowed to do. A user can be assigned multiple TVAuthorization objects in addition to any assigned to groups that the user is a member of.

The administrator must add the `TVAuthorityGroup` attribute to the entry for each user in order for the user to access TransactionVision. This attribute names the full distinguished name of the TransactionVision security group that this user belongs to. The authorities of the security groups are defined under the root TransactionVision directory. Each group can be configured to contain any of the possible permutations.

A TVAuthorization object specifies the following:

- Some combination of named permissions

- A list of allowable projects (if permissions indicate the user can only access certain projects)

- A list of allowable queries (if query permissions are enabled)

- A list of allowable views

Important! To use a user ID instead of user name, you must modify the `LDAP.properties` file to set the UseUID property to true. For more information about this file, see the *TransactionVision Installation and Configuration Guide.*

Default Security Groups

The following four groups are created by default to handle most cases:

- User

- Operator

- Developer

- Admin

The following table shows the TransactionVision operations that can be performed by the default groups:

Task	Admin	Developer	Operator	User
Create, edit, and delete Communication Link templates	Yes	Yes	No	No
Create, edit, and delete Communication Links	Yes	Yes	No	No
Create database schema	Yes	No	No	No

Task	Admin	Developer	Operator	User
Create Analyzer host	Yes	No	No	No
Create, edit, and delete data collection filters	Yes	Yes	No	No
Create, edit, and delete projects	Yes	Yes	No	No
Start and stop projects	Yes	Yes	Yes	No
Create, edit, and delete queries	Yes	Yes	Yes	Yes
View Component Topology, Event Analysis, and Transaction Analysis, and Event Details	Yes	Yes	Yes	Yes
View reports	Yes	Yes	Yes	Yes

TVAuthorization Attributes

The TVAuthorityGroup attribute points to the distinguished name of a TVAuthorization class containing the definitions of privilege for a particular type of user. This class has the following attributes:

Attribute	Description
TVAuthCollectionState	If set to **true**, the user is allowed to start and stop data collection.
TVAuthUserData	If set to **true**, the user is allowed to view user data.
TVAuthModifyProject	If set to **true**, the user is allowed to modify project settings such as communication links and data collection filters.
TVAuthModifyQueries	If set to **true**, the user is allowed to modify and create queries.

Attribute	Description
TVAuthModListedProj	If set to **true** and TVAuthModifyProject is **not** enabled, the user may make changes to a project, but only if it has been entered in the user's project list.
TVAuthOnlyListedQueries	If set to **true**, the user will only be able to use the queries in the list of allowable queries.
TVAuthQueryList	List of queries the user is allowed to use. This attribute may also optionally contain a project name, which limits which projects a user has access to a particular query. The string can be in the form '[projectName + '|'querName'. For example, 'STock|TradeQuery' indicates that a user with these permissions may modify the TradeQuery query, but only in the Stock project.
TVAuthProjectList	List of projects the user is allowed to access.
TVAuthViewList	List of views the user is allowed to access.
TVAuthReportList	List of report groups the user is allowed to access. The value **allGroups** indicates that all reports may be viewed.
TVAuthAccessListed-Projects	If set to **true**, a user may only see projects specified in his project list. If set to **false**, a user has read access to all projects.

You can create a new TVAuthorityGroup, or modify the attribute settings for the default groups.

In the following is an example, the user Joe assigned to the Accounts group, defined by the Accounts object under the root TransactionVision directory (Tvision), as well as the User group. This user has all permissions for both of these groups.

```
+ Users (config file points to 'ou = Users, o = bristol, c = us')
 |——+ Bob
 |    |——+ TVAuthorityGroup = cn=Admin,ou=Security,ou=TVision,o=bristol,c=us
 |——+ Joe
 |    |——+ TVAuthorityGroup = cn=Accounts,ou=Security,ou=TVision,o=bristol,c=us
 |                           = cn=User,ou=Security,ou=TVision,o=bristol,c=us
 |——+ ...
 - - -
+ Tvision
 |——+ Security
 |    |——+ User
 |    |——+ Admin
 |    |——+ Develop
 |    |——+ Accounts
 |    |    |——+ Not allowed to view user data
 |    |    |——+ Not allowed to modify presentation queries
 |    |    |——+ Not allowed to modify data collection
 |    |    |——+ Projects
 |    |    |    |——+ Accounting
 |    |    |——+ Views
 |    |    |    |——+ Reports
 |    |    |    |——+ ...
```

Important! To use a prefix other than "cn" for the user name, modify the `LDAP.properties` file to set the RDNPrefix property to the prefix you wish to use. For more information about this file, see the *TransactionVision Installation and Configuration Guide*.

TVPerson Class

This class is derived from inetOrgPerson, with the TVAuthorityGroup attribute added to it. To use an existing user directory, add the TVAuthorityGroup attribute as an optional attribute to the class that your users are defined as. Otherwise, use the TVPerson class as a base class for any added users.

Using the WebLogic LDAP Server

In general, the embedded WebLogic LDAP server works as other LDAP servers do. However, there are some unique aspects that you should be aware of.

Accessing the WebLogic LDAP Server

Access the embedded LDAP server via the URL of your WebLogic administration server. For example if your administration server runs at `http://mymachine:7001/`, set the `LDAPServerName` property in `Ldap.properties` to `ldap://mymachine:7001/`.

Before running the **TVisionSetup** script, change the generated Credential to your own password to access the server as follows:

1. Expand the domain (for example, mydomain).

2. Select the Security > Embedded LDAP section.

3. Change the credential.

When running **TVisionSetup** to initialize LDAP entries for TransactionVision, use **Admin** as the user name and the changed credential as the password. For more information, see the WebLogic documentation, "Managing the Embedded LDAP Server."

Default Root Object

By default, WebLogic creates a root level object with the name of your domain. Use this value as the root domain in all relevant places in your TransactionVision configuration files.

For example, if your domain is named *mydomain*, you root object name is `dc=mydomain`. If you create your TransactionVision object in the default location, pass that value as the root suffix to the TransactionVision scripts that initialize your LDAP settings. Using this example, set the `UserDirectoryLocation` property in `Ldap.properties` to `ou=Users,ou=TVision,dc=mydomain`.

Modifying Access to the LDAP Server

By default, WebLogic limits access to the embedded LDAP server to users authenticated as the administrator account. For TransactionVision to use the LDAP server, you must change the default settings to grant read access to the TransactionVision LDAP hierarchy. The `<WL_HOME>/server/lib/acls.prop` file controls the access settings.

At minimum, TransactionVision requires the ability to access the `TVAuthorityGroup` attribute of the user logging in and must be able to read all attributes of the `AuthorityGroup` this attribute points to.

To grant read access to everyone, uncomment the default rule that grants read access to everyone.

To grant required permissions only within the TransactionVision tree, add the following entry:

```
ou=TVision,dc=mydomain|subtree#grant:b,s,r,c,t#[all]#public;
```

For more information about modifying access to the LDAP server, see the WebLogic *Managing WebLogic Security* document, located at http://e-docs.bea.com/wls/docs81/secmanage/ldap.html.

Reports Access

The `<TVISION_HOME>/config/ui/Reports.xml` report definition file defines the reports are available to TransactionVision users. It groups reports into report categories.

The following is a sample from the Reports.xml file. It defines a category called "Performance Analysis & Problem Resolution." This category contains a "Reports" subcategory, which contains one or more reports.

```
<ReportCategory name="performance"
  title="Performance Analysis &amp; Problem Resolution">
  <ReportSubCategory name="reports" title="Reports">

...
  </ReportSubCategory>
</ReportCategory>
```

A `ReportCategory` defines a group of reports that are linked to a particular user permission. It has the following attributes:

- `name`
  A ReportCategory *name* corresponds to the name specified in a user's LDAP settings. If this name appears in the user's report LDAP settings then the user will be able to view this group of reports. A user whose list of report groups contains the value of allGroups can see all available reports.

- `title`
  The *title* attribute is the description of this report category and appears in the table header on the reports page.

The `Report` field defines the description, and corresponding link to the JSP that generates/displays a particular report. For more information about creating reports and defining them in the `Reports.xml` file, see the *TransactionVision Programmer's Guide*.

The sample reports that ship with TransactionVision are included in the `tvision.ear`, and will be installed in WebSphere along with the other TransactionVision components by default. They can also be found in your install area under the *ui/ReportSample* directory.

SSL Setup

The following steps set up SSL with SecureWay 3.2 and Java Secure Socket Extensions (JSSE). For different configurations, these steps may vary.

1. Configure the LDAP server to use SSL. First use the SecureWay **ikeyman** key management tool to generate a certificate and keystore for the LDAP server, then enable SSL on the LDAP server and specify the generated key database and certificate. Choose Server Authentication as the SSL authentication method. In the SecureWay administration interface, use Security > SSL > Settings. On platforms without the SecureWay administration tool, edit the configuration files manually. The default SSL port is 636. For more information about configuring the LDAP server to use SSL, see the *IBM LDAP Implementation Cookbook.*

2. Download the Sun JSSE package from http://java.sun.com. This package provides a number of classes required to use SSL. Install these jar files so that they are available in the classpath. Note: if you use a third-party JSSE package, set the CryptProvider property in `Ldap.properties` to a different Cryptograph Service Provider.

3. Set up certificates on the TransactionVision side. Because JSSE uses a different format from the key database created for SecureWay, you must create another key storage. From the IBM key management utility, select Extract Certificate and export the certificate you created for SecureWay using the base 64 encoding option. This creates a file you can import into the **keytool** utility used by JSSE to create certificates. Use the following command to import the certificate, giving it the same name as in the SecureWay definition:

```
keytool -import -alias certName -file file.arm
-keystore keystoreLocation
```

The first time you use this command, you must set a password, which will be required to make further changes to this keystore.

4. Set the following properties in `Ldap.properties`:

Property	Description
UseSSL	Set to true to enable SSL.
CertificateTrustStore	Set to the *keystoreLocation* specified in step 3. This property sets the javax.net.ssl.trustSTore property to point to this file.
LDAPServerName	Set to point to the SSL port defined in step 1.

5. Restart the TransactionVision web user interface using your application server administration console.

Troubleshooting

Communication Link Problems

If your project is not receiving events that you expect it to receive, there may be a problem with the communication link or links you are using. Perform the following steps to diagnose the problem:

1. Ensure that you are not using a data collection filter that prevents Sensors from collecting the expected events and that you are not using a project query that prevents you from seeing the events even though they are in the project database.

2. Choose Current Project > Project Status to verify that you have activated at least one communication link for the project.

3. Choose Current Project > Communication Links, select each activated communication link, pick an Analyzer from the drop down list, and click Test. If the communication link test fails, the error message should indicate which queue or queue manager failed; make sure the specified queue or queue manager has been created.

4. If the communication link test passes, ensure that a Sensor is intercepting WebSphere MQ API calls from the monitored application. On UNIX and Windows platforms, set the TVISION_BANNER environment variable, then start the application. A banner indicating that the application is loading the Sensor should appear. To disable this behavior, unset TVISION_BANNER. This environment variable can be set to any value. On Windows, it must be set to a value other than an empty string. On OS/400, TVISION_BANNER does not display the library path as it does on UNIX.

   If not, set the environment variable for your platform to cause the application to load the Sensor library instead of the standard WebSphere MQ library. The following table shows the appropriate environment variable for each platform:

Platform	Environment Variable
Windows 2000	PATH
Sun Solaris	LD_LIBRARY_PATH
HP-UX	SHLIB_PATH
Linux	LD_LIBRARY_PATH
IBM AIX	LIBPATH
Compaq Tru64	LD_LIBRARY_PATH

On OS/390, the CICS region prints a message when it connects to the queue manager indicating that the WebSphere MQ crossing exit was found and will be used. If the WebSphere MQ crossing exit was not found or loaded, a message indicating the error will appear. If you are not sure how to solve the problem from the error message, contact Bristol Support.

5. If the Sensor library is being loaded by the application, check the application host's system log for any error messages. If there are error messages but you are not sure how to proceed, contact Bristol Support.

6. If there are no error messages in the system log, enable trace logging for the active communication links. Look at the trace messages in the application host's UNIX system log, Windows event log, OS/390 operator console log, or OS/400 user's job log for any WebSphere MQ-related errors or messages that indicate that the data collection criteria such as collection time or collection entities were too restrictive.

On UNIX platforms, you can specify the log facility by setting the TVISION_SYSLOG environment variable to one of the following values: user, local0, local1, local2, local3, local4, local5, local6, or local7. If TVISION_SYSLOG is not set or is set to a value other than those listed, TransactionVision uses local0. The target log file must already exist.

To view the job log on OS/400, use the **DSPJOBLOG** command.

7. If you are still unable to resolve a communication link problem, contact Bristol Support.

Disabling SHLIB_PATH on HP-UX

On HP-UX, you can use the following command to turn off shared library resolution using SHLIB_PATH on an executable:

```
chatr +s disable executable_name
```

The output of `chatr executable_name` will be as follows:

```
shared executable
shared library dynamic path search:
    SHLIB_PATH    disabled  second
    embedded path disabled  first Not Defined
```

In this case, the executable will not be able to load the Sensor. To enable run-time location of the Sensor, you must turn on the loader's use of the SHLIB_PATH with the following command:

```
chatr +s enable executable_name
```

Displaying Perl and Java Program Names

By default, the TransactionVision displays `Perl` or `java.exe` as the program name for Perl or Java applications. To display the program name, set the TVISION_REPORT_ARGS environment variable to any value. to disable this behavior, unset TVISION_REPORT_ARGS. This environment variable can be set on any platform except OS/390. On Windows, it must be set to a value other than an empty string.

Monitoring Triggered Programs

Triggered programs on UNIX platforms can be monitored if two conditions are met:

1. The `runmqtrm` trigger monitor is not setuid/gid, or you are running `runmqtrm` as the mqm user. (This is because the programs started by the setuid/gid `runmqtrm` will ignore any environment variables modifying the path if the user is not the setuid/gid user - the mqm user in this case.)

2. You actually trigger a script instead of your program. The script sets the appropriate environment variable (based on the OS platform) to point to the Sensor and then runs the actual program. For example, on Solaris this script might be:

    ```
    #!/bin/sh
    LD_LIBRARY_PATH=/opt/TVision/lib
    export LD_LIBRARY_PATH
    /usr/local/bin/trigproc "$*"
    ```

Where `/usr/local/bin/trigproc` was the program being triggered. The following table shows the appropriate environment variable and default installation directory for each platform:

Platform	Environment Variable	Default Directory
Windows 2000	PATH	`C:\Program Files\Bristol\Trans-actionVision Sensor\lib`
Sun Solaris	LD_LIBRARY_PATH	`/opt/TVision/lib`
HP-UX	SHLIB_PATH	`/opt/TVision/lib`
Linux	LD_LIBRARY_PATH	`/opt/TVision/lib`
IBM AIX	LIBPATH	`/usr/lpp/TVision/lib`
Compaq Tru64	LD_LIBRARY_PATH	`/opt/Tvision/lib`

Important! The `runmqtrm` trigger monitor program cannot itself be sensed by the Sensor, because it uses internal WebSphere MQ functions which are not compatible with the Sensor. To sense the trigger monitor itself, use either the `amqstrg0.c` trigger monitor or build your own using only public WebSphere MQ APIs.

Monitoring Client Applications

If you are monitoring client applications and notice that the WebSphere MQ listener does not generate TransactionVision events when invoked from inetd, check the user id and group id of the mqm user as follows:

```
> id mqm
uid=nnn(mqm) gid=nnn(mqm)
```

Both the user and the group should be mqm. If not, the user mqm should be added to the group mqm. Alternately, you may change the group id on the listener binary to belong to the group that the user mqm belongs to.

Optimizing Performance

To achieve optimum performance when using TransactionVision, follow these guidelines:

- Install the Analyzer on a computer that is fast enough to keep up with the rate of incoming messages from Sensors. For a slower Analyzer host, increase the maximum queue depth on the event queue used for sending event messages from Sensors to the Analyzer.

- Use data collection filters to restrict event collection to event data absolutely necessary to resolve a particular problem. Be sure to delete the default "Collect All" data collection filter if you define a more restrictive filter, since events that meet **any** filter are collected by Sensors.

- Use the **DB2RunStats** or **OracleRunStats** script to optimize database performance. See the *TransactionVision Installation and Configuration Guide* for complete details.

Missing Events

If it appears that events are missing from a project, first make sure that no data collection filter or project query is preventing the events from being collected or displayed. In the data collection filter, if the event retry timeout is set to a value other than Retry Forever and the timeout is being met, events will be lost.

If the events are still missing, make sure that the WebSphere MQ API call uses a valid hConn parameter. The Sensor uses the hConn value to determine which queue manager to connect to for configuration messages; if this value is invalid, the Sensor cannot check for configuration messages.

Queue Manager Channel Limitations

MaxChannels and MaxActiveChannels queue manager properties have to be set to a reasonable amount for TransactionVision. By default, these properties are not set and there is no limitation. For each active communication link, the TransactionVision Analyzer requires two channels plus one channel for each listening thread defined in the communication link.

If these properties are set too low, the user will see strange behavior such as a partial connection to the communication link. It may at first appear that the Analyzer is connected, but it will report a connection error and put the communication link in a "Trying to Connect" state. This behavior usually points to a problem with these queue manager properties not being set properly.

Servlet and EJB Sensor Problems

If you are not able to collect servlet or EJB events from your web application, check the following:

1. You have run **SensorSetup.sh(bat)** to install TransactionVision Servlet Sensor on the application server your web application is running on. The Application server name is case sensitive. "Default Server" is different from "default server" or "DefaultServer" (space counts). Run **SensorSetup.sh(bat)** again if needed.

2. Check config/sensor/SensorConfiguration.xml from your TransactionVision home to make sure the information you provided there is correct and the queue manager is running. Run **SensorSetup.sh(bat)** to modify the setting if needed.

3. Check sensor.log in your log file directory to see if there are any errors.

4. Check WebSphere's system.out and system.err log files to see if there are any errors.

JMS Sensor Problems

If you are not able to collect JMS events from stand-alone JMS application, check the following:

1. Make sure that file java\lib\com.ibm.mqjms.jar is in your TransactionVision home. Run **SensorSetup.sh(bat)** again if it is not there.

2. Your ClassPath points to java\lib\com.ibm.mqjms.jar in TransactionVision home AHEAD of the same file from WebSphere MQ directory.

3. Check config/sensor/SensorConfiguration.xml from your TransactionVision home to make sure the information you provided there is correct and the queue manager is running. Run **SensorSetup.sh(bat)** to modify the setting if needed.

4. If JMS functions are called from servlets, make sure you answer "YES" to the question "Do you wish to monitor JMS methods along with Servlet methods?" when running **SensorSetup.sh(bat)**

Web User Interface Problems

Manually Deleting a Schema

If you manually delete a project's schema (using **CreateSqlScript**), you may still login to TransactionVision, but most pages will not work because the database table does not exist. To remedy this situation, select the database schema associated with the project on the Database Schemas page and click Create Tables.

Loading Actuate Reports

The Actuate report engine requires an X connection to run. Your web server should be started in an environment where DISPLAY is set to a valid X server. By default, it will try to connect to the default display, "0:0". If X is not available or permission does not exist for the user to access X on the server machine, then you must set the DISPLAY environment variable to a valid X server.

If an X server is not available, the Actuate report engine fails to load and TransactionVision displays a message that the page cannot be displayed.

Application Problems

TransactionVision does not require changes to monitored applications. However, applications on the AIX platform may fail after installing a WebSphere MQ support pac. In WebSphere MQ support pacs, internal symbols exported from the TransactionVision WebSphere MQ Sensor on the AIX platform may change. When an internal symbol that has been exported from the Sensor library is no longer available in the WebSphere MQ library, the application cannot start and fails with various symbol resolution errors.

To work around this problem, run the **rebind_sensor** script whenever a WebSphere MQ support pac that modifies the WebSphere MQ libraries (libmqm.a, libmqic.a, libmqm_r.a, libmqic_r.a) are modified. For more information about this script, see Appendix A, "Utilities Reference."

Analyzer Problems

There are several reasons why the Analyzer might fail to process events. The most common are:

- Database errors, such as the database is down for a period longer than the recoonect setting or storage space is full

- Errors caused by custom analysis beans

TransactionVision provides two methods for minimizing the impact if the Analyzer fails to process events for any reason:

• Placing event on the dead letter queue

• Starting recovery mode for the Analyzer

The Dead Letter Queue

If the Analyzer fails for any reason, TransactionVision will not lose any analysis data. Instead, failed events are placed on the dead letter queue defined in the communication link from which the events have been retrieved. See page 74 for instructions on setting the dead letter queue when you create a communication link.

The number of events to be placed on the dead letter queue depends on the value of the batch_commit_count property in the `Analyzer.properties` file. See Appendix C in the *TransactionVision Installation and Configuration Guide* for more information about this file.

Once the cause of the failure is resolved, use the **GetPut** utility to replay the events on the dead letter queue to the event queue, as in the following example:

```
java com.bristol.tvision.admin.GetPut -gq SYSTEM.DEAD.
LETTER.QUEUE -pq TVISION.EVENT.QUEUE -qm YOUR_QUEUE_
MANAGER
```

For more information about this utility, see "GetPut" on page 188.

If the dead letter queue contains events that have been failed **after** the XML data of the event has been written to the database (in the event analysis stage), those events will cause a "Duplicate XML event" warning when they are replayed to the event queue. You can safely ignore this warning.

Analyzer Recovery Mode

The Analyzer uses in-memory caches to accumulate and store statistics data for the Static Topology Analysis view. This data is flushed and written to disk frequently by a separate flush thread. The flushing interval is specified by the flush_interval property in the `StatisticsCache.properties` file. See Appendix C in the *TransactionVision Installation and Configuration Guide* for more information about this file.

While the Analyzer is running,it keeps track of which events the statistics data has been flushed for. If the writing of the cache data fails for any reason, the Static Topology Analysis view statistics will be incomplete, and the project schema will be marked for RECOVERY mode. Failures that can lead to RECOVERY mode include database errors, event processing errors, and abnormal termination of the Analyzer process.

When a project whose schema has been marked for RECOVERY is started, the Analyzer goes into RECOVERY mode to regenerate the statistics for the Component Topology Analysis view, as well as any other missing analysis data, from the stored XML event. The time needed for the recovery process is proportional to the number of events that need to re-analyzed, and is usually slower than the normal event processing rate. A log entry in the `Analyzer.log` file will show how many events need to be recovered.

If TRACE Logging is configured for the Analyzer, the progress of the recover process is also logged. To enable trace logging for the TransactionVision Analyzer, set the value of the `trace` property in the `Analyzer.properties` file to `on`. For more information about this file, see Appendix C in the *TransactionVision Installation and Configuration Guide*.

Once the data has been flushed to disk successfully, the recovery mode of the schema will be reset, and the project will start up normally the next time.

Important! If the statistics data could not be flushed because of a processing error which cannot be resolved, the Analyzer will not be able ro re-analyze the corresponding events and will continut to start up in RECOVERY mode. In such a case, the recovery state can be cleaned out with the **FlushStatusUtil** utility. After the state has been cleanded, the Analyzer will not attempt to recover those events again. For more information about this utility, see "FlushStatusUtil" on page 187.

Analyzer States

You can view Analyzer status in either of the following ways:

- Choose Administration > Analyzers to display the Analyzers page.

- Choose Current Project > Project Status to display the Project Status Page

The following table describes the status values:

Status	Description
COLLECTING EVENTS	The Analyzer is started and collecting events.
FAIL	TransactionVision cannot connect to the Analyzer; an error message is provided.
RECOVERY	The Analyzer is performing statistics recovery. This can happen when the Analyzer has been stopped abnormally (for example, the process has been killed), and not all event statistics have been flushed to disk. In this case the Analyzer will reprocess the last events to get the statistics up-to-date before it starts the normal processing.
QUIESCING	The Analyzer is shutting down all collection threads in response to a stop command. Once all collection threads are shut down, the status changes to NOT COLLECTING EVENTS.
NOT COLLECTING EVENTS	The Analyzer has been stopped and is not collecting events.

Event and Transaction Correlation Problems

The Analyzer issues the following warnings when it detects potential problems correlating events into local transactions or correlating local transactions into business transactions.

Event Correlation

```
TransactionVision Warning(): Number of correlated events
exceeds threshold: event ID='(1245,23)', count=30.
```

The event with the specified ID has been correlated into a message path together with an unusually large number of related events. If you are using custom correlation beans, this warning could indicate that the logic in the custom beans does not work correctly.

If no custom correlation logic is in place, this warning could indicate that the message data in the events used for correlation is not well-suited for the standard correlation, and custom correlation may be necessary. For example, applications may re-use WebSphere MQ message IDs.

Transaction Correlation

```
TransactionVision Warning(): Number of local transactions
in one business transaction exceeds threshold: business
txn ID='32', count=1000.
```

The business transaction with the specified ID consists of an unusually large number of local transactions. If you are using custom analysis beans, this warning could indicate that the logic in the custom beans does not work correctly.

If no custom analysis logic is in place, this warning could indicate that the nature of the message data might not be well-suited for the standard analysis algorithms, and custom correlation/transaction logic may be necessary.

Event Queue Cleanup

There may be times when messages that are not being collected by a TransactionVision project accumulate on the event queue. These events may be targeted for invalid schemas or communication links, or they may be erroneous messages. In either case, you may use the **EventQueueCleanup** utility to remove or replay these messages.

To view a snapshot of the messages on an event queue (TVISION.EVENT.QUEUE on queue manager QMGR1 in this example), enter the following command:

```
EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE -verbose
```

For complete information about **EventQueueCleanup**, see "EventQueueCleanup" on page 184.

Invalid Schema and Communication Link IDs

Events are targeted for a specific database, schema, and communication link identified within TransactionVision by a unique ID. If a schema is deleted or a communication link is changed, the unique ID no longer exists, and any events still on the event queue that targeted for that schema or communication link are no longer valid.

To display all schema and commlink IDs, enter the following command:

```
EventQueueCleanup -showids
```

To delete events targeted for database schemas or communication links which no longer exist from the queue TVISION.EVENT.QUEUE on the queue manager QMGR1, enter the following command:

```
EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE -delete
```

You can also use **EventQueueCleanup** to replay events targeted for invalid schema or communication link IDs to a new schema and communication link so they can be processed by the Analyzer:

```
EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE -replay 2 11
```

In this example, the schema ID for the new schema is 2 and the communication link ID for the new communication link is 11.

Erroneous Messages

Erroneous messages are messages on the event queue that do not appear to be TransactionVision event messages. Erroneous messages could include messages generated from other applications or WebSphere MQ COA report messages used by communication link time skew tests.

To remove erroneous messages from TVISION.EVENT.QUEUE on queue manager QMGR1, enter the following command:

```
EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE -erroneous
```

Events Not Pulled from Event Queue

Event queues can be shared across Analyzers, projects, and communication links. Therefore, events have to be retrieved from the queue based on a specific message ID (correlation ID in the case of Java Sensor events).

The following diagram shows is the layout of the message ID (correlation ID in the case of Java Sensor events) on TransactionVision events:

12 Bytes Database Name	4 Bytes Database Host IP	4 Bytes CommLink ID	4 Bytes Schema ID

Given an assigned communication link, an Analyzer will get events only by a specific message ID/correlation ID. Therefore, it is possible to have events on the event queue which may never be removed from the queue. Different scenerios can cause this.

Dropped Database

If a database is dropped and never created again, events intended for projects that were previously on the dropped database may exist on the event queue.

Deleted Schema

If a project schema is deleted, events intended for the schema may exist on the event queue. Recreating the schema will not reassign these events on the event queue to the newly created schema because a new unique ID is always assigned to a created schema, even if it is given the same name as the previous.

Deleted or Changed Commlink

If a communication link is deleted from a project or deassigned from the Analyzer in a project, events intended for that communication link may exist on the event queue. If the communication link has just been deassigned, simply reassigning it to the Analyzer will allow the Analyzer to retrieve the pending events on the event queue. However, if the communication link has been deleted from the project, recreating the communication link (even if created from a communication link template) will not provide the means to retrieve the events left on the event queue. This is because every time a communication link is created, it is assigned a unique ID. In addition, if a communication link is changed, a new unique id is assigned to it, and old events will remain on the event queue.

WebSphere MQ Anomalies

TransactionVision events are not the only kinds of messages that may appear on a event queue.

TransactionVision uses WebSphere MQ Confirmation of Arrival (COA) messages to determine time skew across communication links. This means that a COA request message is sent to the configuration queue, and the COA report is sent back on the event queue.

If a remote (using multiple queue managers) communication link is being used, the COA report has to travel across a WebSphere MQ channel to reach the final event queue destination. If there are problems in the WebSphere MQ environment (for example, the channel is down), the Analyzer may timeout waiting for a COA report message. If the WebSphere MQ problem is later fixed, the COA report message may finally arrive in the event queue. This COA report message will never be removed from the event queue, because the analyzer has given up on it.

There is also the possibility of other WebSphere MQ applications maliciously or accidently putting messages on the TransactionVision event queue. The Analyzer simply ignores these because they will most likely not have a matching message ID/correlation ID it is looking for, and therefore will remain on the event queue indefinitely.

Problems Importing System Tables into an Oracle Database

The referential constraints of TransactionVision system tables will prevent Oracle data import utility from loading data into existing tables. The Oracle data import utility returns an error similar to the following example:

```
IMP-00019: row rejected due to ORACLE error 2291
IMP-00003: ORACLE error 2291 encountered
ORA-02291: integrity constraint (TVISION.FK1_FILTER)
violated - parent key not found
Column 1 17
Column 2 Collect All
Column 3 16
Column 4
00540001020C0000000100000001000000179F0700008ADC00...
0 rows imported
```

If you encounter this error, disable the constraints, import the data, and then re-enable the constraints.

For more information, see the following Oracle document:
http://download-west.oracle.com/docs/cd/A91202_01/901_doc/server.901/a90192/ch02.htm#1005094

Disabling Referential Contraints

To disable the referential contraints, use the following SQL commands:

```
ALTER TABLE TVISION.ANALYZER_PROJ_COMMLINK DISABLE
CONSTRAINT FK2_ANLZ_PROJ_LINK;

ALTER TABLE TVISION.ANALYZER_SCHEMA DISABLE CONSTRAINT
FK2_ANLZ_SCHEMA;

ALTER TABLE TVISION.COMMLINK_FILTER DISABLE CONSTRAINT
FK1_LINK_FILTER;
```

```
ALTER TABLE TVISION.COMMLINK_FILTER DISABLE CONSTRAINT
FK2_LINK_FILTER;

ALTER TABLE TVISION.FILTER DISABLE CONSTRAINT FK1_FILTER;

ALTER TABLE TVISION.PROJECT DISABLE CONSTRAINT
FK1_PROJECT;

ALTER TABLE TVISION.PROJECT_COMMLINK DISABLE CONSTRAINT
FK1_PROJ_LINK;

ALTER TABLE TVISION.QUERY DISABLE CONSTRAINT FK1_QUERY;
```

Re-enabling Referential Contraints

To re-enable the referential constraints after importing the data, use the following SQL commands:

```
ALTER TABLE TVISION.ANALYZER_PROJ_COMMLINK ENABLE
CONSTRAINT FK2_ANLZ_PROJ_LINK;

ALTER TABLE TVISION.ANALYZER_SCHEMA ENABLE CONSTRAINT
FK2_ANLZ_SCHEMA;

ALTER TABLE TVISION.COMMLINK_FILTER ENABLE CONSTRAINT
FK1_LINK_FILTER;

ALTER TABLE TVISION.COMMLINK_FILTER ENABLE CONSTRAINT
FK2_LINK_FILTER;

ALTER TABLE TVISION.FILTER ENABLE CONSTRAINT FK1_FILTER;

ALTER TABLE TVISION.PROJECT ENABLE CONSTRAINT FK1_PROJECT;

ALTER TABLE TVISION.PROJECT_COMMLINK ENABLE CONSTRAINT
FK1_PROJ_LINK;

ALTER TABLE TVISION.QUERY ENABLE CONSTRAINT FK1_QUERY;
```

Extending TransactionVision Sensors

The Sensor library provides two types of extension APIs. One provides custom trace settings to the TransactionVision views. The other enables you to implement user exits for each WebSphere MQ API called by the Sensor.

Custom Trace Settings

The **BTTRACE** function can be called by applications to provide custom trace messages to the TransactionVision analysis views. These messages appear in the Event Analysis and Transaction Analysis views. The calling application must link against the Sensor library to use this facility. On OS/390 CICS, the application program must be link-edited with the Sensor CICS stub program SLMCSTUB. The sample in the Sensor samples directory or library demonstrates custom trace messages as well as how to link your application against the Sensor. It also includes a project file and makefiles for supported platforms.

To implement custom tracing, the application program must call **BTGETHMQ** to get a **BTTRACE** handle, then call **BTTRACE** once for each message to be logged, and then call **BTRELH** to release the handle when all tracing by that program is completed. A sample C program (SLMTSAMP) and header file (BTTRACE) are provided in the samples directory or library.

BTGETHMQ — Get BTTRACE Handle

The **BTGETHMQ** call initializes **BTTRACE** operations and provides a handle, which is used by the application on subsequent **BTTRACE** calls.

Synopsis

```
BTGETHMQ(Hconn, BTHandle)
```

Parameters

`Hconn` (MQHCONN) - input

Connection handle representing the connection to the WebSphere MQ queue manager. This handle is provided by the **MQCONN** or **MQCONNX** calls. ON OS/390 CICS and OS/400 platforms, this parameter is ignored.

`BTHandle` (BTHANDLE) - output

The **BTTRACE** handle, which must be specified on all subsequent calls to **BTTRACE**.

C Language Invocation

```
BTGETHMQ(Hconn, &BTHandle);
```

Declare the parameters as follows:

```
MQHCONN Hconn        /* Connection handle */
BTHANDLE BTHandle    /* BTTRACE handle */
```

COBOL Language Invocation (OS/390)

```
CALL BTGETHMQ USING HCONN, BTHANDLE.
```

Declare the parameters as follows:

```
**   Connection handle
01   HCONN PIC S9(9) BINARY.
**   BTTRACE handle
01   BTHANDLE PIC S9(9) BINARY.
```

PL/1 Language Invocation (OS/390)

```
call BTGETHMQ(Hconn, BTHandle);
```

Declare the parameters as follows:

```
dcl Hconn fixed bin(31);    /* Connection handle */
dcl BTHandle fixed bin(31); /* BTTRACE handle */
```

System/390 Assembler-language Invocation (OS/390 Only)

```
CALL BTGETHMQ,(HCONN,BTHANDLE)
```

Declare the parameters as follows:

```
HCONN        DS   F        Connection handle
BTHANDLE     DS   F        BTTRACE handle
```

BTTRACE — Log Trace Message

The **BTTRACE** call logs a trace message to be sent to the TransactionVision views.

Synopsis

```
BTTRACE(BTHandle, Severity, MsgTypeSize, MsgType, MsgSize,
Message)
```

Parameters

BTHandle (BTHANDLE) - input
> The **BTTRACE** handle returned by the **BTGETHMQ** call.

Severity (32-bit unsigned integer) - input
> A message severity level that may be used to filter trace events. The following severity levels and values are defined:
>
> TVISION_SEVERITY_CRITICAL 0
> TVISION_SEVERITY_ERROR   1
> TVISION_SEVERITY_WARNING 2
> TVISION_SEVERITY_INFO   3
> TVISION_SEVERITY_DEBUG   4

MsgTypeSize (32-bit unsigned integer) - input
> The length of the MsgType parameter.

MsgType (character) - input
> A message type composed of any text, which may be used to filter trace events.

MsgSize (32-bit unsigned integer) - input
> The length of the Message parameter.

Message (character) - input
> The text of the message to be logged.

C Language Invocation

```
BTTRACE(BTHandle, Severity, nMsgTypeSize, pchMsgType,
nMsgSize, pchMessage);
```

Declare the parameters as follows:

```
BTHANDLE BTHandle          /* BTTRACE handle */
int Severity               /* Message severity */
unsigned int nMsgTypeSize  /* Message type size */
const char * pchMsgType    /* Message type */
unsigned int nMsgSize      /* Message size */
const char * pchMessage    /* Trace message text */
```

COBOL Language Invocation (OS/390)

```
CALL BTTRACE USING BTHANDLE, SEVERITY, MSGTYPESIZE,
MSGTYPE, MSGSIZE, MESSAGE.
```

Declare the parameters as follows:

```
**   BTTRACE handle
01   BTHANDLE PIC S9(9) BINARY.
**   Message severity
01   SEVERITY PIC S9() BINARY.
**   Message type size
01   MSGTYPESIZE PIC S9(9) BINARY.
**   Message type
01   MSGTYPE PIC X(n).
**   Message size
01   MSGSIZE PIC S9(9) BINARY.
**   Trace message text
01   MESSAGE PIC X(n).
```

PL/1 Language Invocation (OS/390)

```
call BTTRACE(BTHandle, Severity, MsgTypeSize, MsgType,
MsgSize, Message);
```

Declare the parameters as follows:

```
dcl BTHandle fixed bin(31);     /* BTTRACE handle */
dcl Severity fixed bin(31);     /* Message severity */
dcl MsgTypeSize fixed bin(31)   /* Message type size */
dcl MsgType char(n);            /* Message type */
dcl MsgSize fixed bin(31);      /* Message size */
dcl Message char(n);            /* Trace message text */
```

System/390 Assembler-language Invocation (OS/390 Only)

```
CALL BTTRACE,(BTHANDLE,SEVERITY,MSGTYPESIZE,MSGTYPE,  X
      MSGSIZE,MESSAGE)
```

Declare the parameters as follows:

```
BTHANDLE    DS  F       BTTRACE handle
SEVERITY    DS  F       Message severity
MSGTYPESIZE DS  F       Message type size
MSGTYPE     DS  CL(n)   Message type
MSGSIZE     DS  F       Message size
MESSAGE     DS  CL(n)   Trace message text
```

BTRELH — Release BTTRACE Handle

The **BTRELH** call releases the **BTTRACE** handle. This function should be called when the application has completed tracing.

Synopsis

```
BTRELH (BTHandle)
```

Parameters

BTHandle (BTHANDLE) - input
    The **BTTRACE** handle to be released.

C Language Invocation

```
BTRELH(BTHandle);
```

Declare the parameter as follows:

```
BTHANDLE BTHandle   /* BTTRACE handle */
```

COBOL Language Invocation (OS/390)

```
CALL BTRELH USING BTHANDLE.
```

Declare the parameter as follows:

```
**  BTTRACE handle
01  BTHANDLE PIC S9(9) BINARY.
```

PL/1 Language Invocation (OS/390)

```
call BTRELH(BTHandle);
```

Declare the parameter as follows:

```
dcl BTHandle fixed bin(31); /* BTTRACE handle */
```

System/390 Assembler-language Invocation (OS/390 Only)

```
CALL BTRELH,(BTHANDLE)
```

Declare the parameter as follows:

```
BTHANDLE    DS  F       BTTRACE handle
```

User Exits for WebSphere MQ APIs

During typical TransactionVision operation, WebSphere MQ APIs called by an application are intercepted by the Sensor, which collects event information about the API and then forwards the API to the WebSphere MQ library. However, you may configure the Sensor library to call a pre-defined API for each WebSphere MQ API on exit. These exit APIs will have access to parameter information from the API as well as the message buffer (if applicable).

An example application of user exits would be to encrypt the data on each end of the message flow without interfering with the application code itself.

Important! User exits are supported on all platforms except IBM OS/390.

User exits are implemented in the `btmqexit` library. If this shared library exists in the library search path, the Sensor will load it dynamically, enabling you to implement user exits without changing application source code.

To implement user exits for the Sensor implementation of WebSphere MQ APIs, perform the following steps:

1. Implement desired exit behavior in `btmqexit` source code.

2. Build the `btmqexit` library.

3. Add `btmqexit` to your library search path.

Important! The OS/400 platform does not support dynamic library loading with **dlopen()**, so user exits are supported in a different manner. On the OS/400 platform, perform the following steps to implement user exits for WebSphere MQ APIs:

1. Set the environment variable TVISION_USEREXIT to true to use the TransactionVision crossing exit.

2. Create the service program BTMQEXIT using the binder source file /SRC/BTM-QEXITBN provided with TransactionVision.

3. Add the library containing your BTMQEXIT service program in front of the TransactionVision library in your library list.

Implementing Exit Behavior

The `btmqexit` library implements a pre-defined API associated with each WebSphere MQ API. The following table shows the relationship between these APIs:

WebSphere MQ API	Associated btmqexit API
MQBACK	BTBACK
MQBEGIN	BTBEGIN
MQCLOSE	BTCLOSE
MQCMIT	BTCMIT
MQCONN	BTCONN
MQCONNX	BTCONNX
MQDISC	BTDISC
MQGET	BTGET
MQINQ	BTINQ
MQOPEN	BTOPEN
MQPUT	BTPUT
MQPUT1	BTPUT1
MQSET	BTSET

The following files are provided with your TransactionVision installation:

- `btmqexit.c`, located in `<install_directory>/TVision/src`

- `btmqexit.h`, located in `<install_directory>/TVision/include`

- `btmqexit.exp`, located in `<install_directory>/TVision/src`, is an export file needed to create the `btmqexit` library on the AIX platform.

btmqexit.c provides a stub implementation of each pre-defined API. Copy this file and modify it to implement the desired action for each API. In the example located in <install_directory>/TVision/samples/btmqexit, the code for each API has been modified as follows to print the API completion code and reason code upon return of each WebSphere MQ API:

```
static void BTCONN(BTCMQA *pCMQA, PMQCHAR pQMgrName,
    PMQHCONN pHconn, PMQLONG pCompCode, PMQLONG pReason)
{
    if (!pCMQA || !pCMQA->pXPB)
        return;

    if (pCMQA->pXPB->ExitReason == MQXR_BEFORE)
    {
        fprintf(stderr,"BTMQEXIT: Calling MQCONN\n");
    }
    else
    {
        if (pCompCode && pReason)
            fprintf(stderr,"BTMQEXIT: MQCONN Returned:
Completion Code: %ld "
                "Reason Code: %ld\n",*pCompCode,*pReason);

    }
    pCMQA->pXPB->ExitResponse = MQXCC_OK;
}
```

Important! When implementing user exits, note that you annot link with WebSphere MQ or the Sensor on any platforms because of the potential for recursive exit calls.

Building the Exit Library

The btmqexit sample directory also contains sample project files and makefiles for building the sample library on supported platforms. The following table shows the target library name for each platform:

Platform	Target Library Name
HP-UX	libbtmqexit.sl
IBM AIX	libbtmqexit.a
Solaris	libbtmqexit.so

Platform	Target Library Name
Windows 2000	`BTMQEXIT.DLL`

Adding the Library to the Library Search Path

For the Sensor to load the `btmqexit` library, you must add its directory location to an environment variable setting on the computer where the monitored application runs before starting the application. The following table shows the appropriate environment variable to set for each distributed platform.

Platform	Environment Variable
Windows 2000	PATH
Sun Solaris	LD_LIBRARY_PATH
HP-UX	SHLIB_PATH
IBM AIX	LIBPATH
RedHat Linux	LD_LIBRARY_PATH
Compaq Tru64	LD_LIBRARY_PATH

Utilities Reference

CreateLdapEntries

Location:

> TVISION_HOME/bin/CreateLdapEntries.[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.

Syntax:

> CreateLdapEntries [-h] [-u] [-n<dn>] [-p<passwd>]
> [-a<auth>] [-d<dn>] [-c<dn>]

Options:

Option	Description
-h	Print the usage message.
-u	Create a default administration user under the tvision group location.
-n<dn>	Use <dn> as the distinguished name for authentication.
-p<passwd>	Use <passwd> as the password for authentication.
-d<dn>	Create the default tvision group at this location.

Option	Description
-a\<auth\>	Use \<auth\> as the authentication mechanism. The default is "simple" if \<dn\> is specified; otherwise, "none".
-c\<dn\>	Create the default root suffix (for example, co=bti,c=us).

Examples:

Create authentication with name as "cn=root" and password as "secret". Create default tvision group as "ou=TVision" at location "o=bti;c=us" and create a default administration user under the tvision group location:

```
CreateLdapEntries[.bat|.sh] -u "-ncn=root" -psecret
"-do=bti,c=us"
```

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 a 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.

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
`-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. See "Notes" for information about using this option with an Oracle database.
`-dbproperties (-db) FILE`	Use Database.properties file FILE.
`-import (-i)`	Generates a database import script. To run database scripts, use the command `db2 -n -t -f <sql script filename>`. For a DB2 database, you can combine this option with the `-fileType` and `-lobPath` options to customize the data format and the location of LOB. You may NOT combine this option with the `-noscript` or `-execute` options.
`-export (-ex)`	Generates a database export script. To run database scripts, use the command `db2 -n -t -f <sql script filename>`. For a DB2 database, you can combine this option with the `-fileType` and `-lobPath` options to customize the data format and the location of LOB. You may NOT combine this option with the `-noscript` or `-execute` options.

Option	Description
`-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 of 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
   ```

Notes:

For the Oracle database, when using CreateSqlScript script, if a different tablespace is specified (using -ts), the quota is not set for the specified tablespace. To workaround this, follow the steps below:

1. Generate the SQL file, `create_project_tables.sql`, using the CreateSqlScript program, without using the -e option.

2. Modify the SQL statements in `create_project_tables.sql` from:

CREATE USER "TEST" PROFILE "DEFAULT" IDENTIFIED BY "TEST"
DEFAULT TABLESPACE "TESTTBS" TEMPORARY TABLESPACE "TEMP"
QUOTA UNLIMITED ON USERS ACCOUNT LOCK;

to:

CREATE USER "TEST" PROFILE "DEFAULT" IDENTIFIED BY "TEST"
DEFAULT TABLESPACE "TESTTBS" TEMPORARY TABLESPACE "TEMP"
QUOTA UNLIMITED ON TESTTBS ACCOUNT LOCK;

where TESTTBS should be changed to the tablespace you are using, TEST should be changed to the schema being created.

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 **cron** 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 a DB2 database instance 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 DB2 installation location and the correct TVISION_HOME location.

Syntax:

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

Options:

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
-v7	Use in a DB2 7.x environment. The default operatin is for DB2 8.1.

DeleteEvents

Location:

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

Purpose:

Allow the user to delete events in EVENT tables and clean up contents in other project tables related to deleted events.

The utility (by default) does only delete the event data. In order to clean up other related table data like system model objects and transaction data, use the options -deletePiis and -deleteTxns described below. If you have deleted event data without these options, you can clean up the remaining data by running the DeleteEvents utility again with the option -cleanup.

Syntax:

```
DeleteEvents
-query queryName -project projName
  [-deletePiis] [-deleteTxns] [-interationcount EVENTS]
  [-commit INTERVAL] [-timeout MINUTES][-force]
  [loglevel LOGLEVEL]
|-from TIME -to TIME | -older TIME | -newer TIME] [-minage HOURS]
  [-schema SCHEMA | -project projName]
  [-timezone TZID][-businesstxns[-state STATE][-result RESULT]]
  [-deletePiis] [-deleteTxns]
  [-deleteStats [topology][business]][-commit INTERVAL]
  [-timeout MINUTES] [-interationcount EVENTS] [-force]
  [-loglevel LOGLEVEL]
|-cleanup {[-sysmodel][-transactions]
  [-commit INTERVAL][-timeout MINUTES][-force]
  [-loglevel LOGLEVEL][-threadcount THREADS] [-nosplit]
```

Options:

Option	Description
-project (-p) projName	Specifies the project name.
-query (-q) queryName	Specifies the name of the query.
-commit(-c) INTERVAL	Commit database transactions after the specified INTERVAL. The default interval is 1000.
-from(-f) STARTTIME	Specifies the start time (Time format = mm/dd/yyyy hh:mm:ss.SSS, event time in the time zone you set (your local time zone by default) or, for relative time, n[hr|min])
-to(-t) ENDTIME	Specifies the end time (Time format = mm/dd/yyyy hh:mm:ss.SSS, event time (your local time zone by default) or, for relative time, n[hr|min])
-older (-o) TIME	Deletes events older than the specified time.
-newer (-n) TIME	Deletes events newer than the specified time.

Option	Description
-minage (-m) HOURS	By default, the **DeleteEvents** utility does not allow deleting events from the last 48 hours. Use this option to override the default. No events from the last number of hours specified by the HOURS value will be deleted.
-schema(-s) SCHEMA	Database schema for where events are stored.
-businesstxns (-bt) [-state STATE] [-result RESULT]	Deletes only events of transactions with the specified transaction state or result. State values: -1=UNKNOWN 0=PROCESSING 1=COMPLETED Result values: -1=UNKNOWN 0=FAILED 1=SUCCESS
-deleteStats (-ds) [topology(t)] [business(b)]	Deletes statistics data from the topology or business transaction statistics table. This is optional. The time period specifies the time slices for which statistics are deleted.
-timezone (-z) TZID	Timezone ID (for example, GMT, EST, etc.)
-help(-h)	Displays the usage message.
-deletePiis (-dp)	Delete program instance objects from the system model table together with the events (DEFAULT: off).
-deleteTxns (-dt)	Delete transactions from the transaction tables together with the events (DEFAULT: off).

Option	Description
`-cleanup (-cl)` `-sysmodel (-sys)` `-transactions (-txns)`	Cleans up the system model tables (removes all program instance objects which are no longer referenced) or transaction tables (removes all transaction objects which are no longer referenced). You must specify either system model tables or transactions.
`-interactioncount (-it) EVENTS`	Maximum number of events to process before cleaning out the system model and transaction tables. The default is 1000.
`-timeout (-ti) MINUTES`	Force deletion to end after the specified time. The default is 60 minutes.
`-force (-fc)`	Force deletion to stop immediately after a cancel or timeout. This option is always is effect in timestamp deletion mode (for example, with the `-older` option).
`-loglevel (-log)` `LOGLEVEL`	Log output level. The default is 0.
`-threadcount (-tc)` `THREADS`	Number of deletion threads (only applicable if using -older).
`-nosplit`	Disables deletion of the same table by multiple concurrent threads.

Notes:

Modes

This utility operates in three modes: deleting by query mode, deleting by time mode, and cleaning up tables. Therefore, the options are grouped into three sets.

- Deleting by time options include: -from, -to, -older, -newer, -timezone, -deletePiis, -deleteTxns, -deleteStats, -businesstxns, -state, -result (and -threadcount, -nosplit if using -older). The -force option is always in effect when deleting by time.

- Deleting by query options include: -query, -deletePiis, -deleteTxns

- The cleanup mode options include: -cleanup, -sysmodel, -transactions

Options that can be used in all three modes include: -c, -ti, -fc, -log. Otherwise, these sets of options may not be mixed. For example, the following commands are INVALID and will result in errors:

```
DeleteEvents -project MYPROJECT -query MYQUERY -business
DeleteEvents -project MYPROJECT -query MYQUERY -older
05/15/2003 03:00:00.000
```

Deleting Events While the Analyzer is Running

Execution of the **DeleteEvents** utility while the Analyzer is processing events is not recommended. However, sometimes it may not be feasible to shut down the Analyzer while **DeleteEvents** is running. In that case, make sure that you do **not** delete transactions for which events are likely to come in.

On many systems, deleting events from 2 days to a week in the past can ensure this scenario does not occur. However, you need to consider the duration of your business transactions to determine a period beyond which events can be deleted. For example, if your business transactions can span a day, do not delete events in the last day, if your business transactions can span a week, do not delete events in the last week while the Analyzer is running. The utility will (by default) not delete any events that are younger than 48 hours. If you are sure that no transaction duration will ever exceed a certain value, you can override this behavior by specifying the option -minage *n*, where *n* is the time buffer to exclude in hours.

Performance

Deleting by time mode provides better performance results than deleting by query. The performance of the deletion options can be ordered in the following way:

1. Deletion by query (low performance)

2. Deletion by time with options -from/to or -newer (medium performance)

3. Deletion by time with option -older (high performance)

The deletion by time with option -older is able to take advantage of timestamp columns present in the tables and achieves much higher performance than the other methods. Also, it can execute the deletion process with multiple threads (option -threadcount), where each thread will delete different portions of the tables at the same time. The default number of threads is 4. You should adjust this option to the number of CPUs (or higher) in your system.

The following conditions may also affect the performance of the **DeleteEvents** utility:

- Make sure your database log space is sufficient, and adjust the number of uncommitted database operations (option -commit) accordingly; otherwise, the deletion process will fail running out of logspace. Usually a larger commit count will yield better deletion performance.

- On DB2, you could encounter deadlocks if the deletion is using multiple threads and your database lock storage is low. In that case, please adjust the value of LOCKLIST and MAXLOCKS accordingly. On Oracle this is of no concern because Oracle will never escalate row locks to table locks. Alternately, use the option -nosplit to DeleteEvents. This disables the concurrent deletion of a single table by multiple threads.

Time Formats

Time formats may be absolute format or relative format. Absolute times are specified as mm/dd/yyyy hh:mm:ss.SSS. Relative times are specified as n[hr|min]. Relative time is offset to the time you run the script. For example "12 hr" means "12 hours before the current time." You can specify the time in any of the following ways. The descriptions assume that the script was run at 9:00 a.m.

-f 2 hr -t 1 hr	Deletes events from 7:00 a.m. to 8:00 a.m.
-o 1 hr	Deletes events that came into the system before 8:00 a.m.
-n 1 hr	Deletes events within the last hour.

Examples:

1. Delete everything (event and related data) from project PROJECT which is older than 72 hours using 2 threads and commit database transactions after 10000 operations:

   ```
   DeleteEvents -project PROJECT -older 72 hr -dp -dt
   -ds b t -tc 2 -c 10000
   ```

2. Delete events from project PROJECT using query QUERY and commit the database transaction after deletion of every 10 events:

   ```
   DeleteEvents –project PROJECT –query QUERY –c 10
   ```

3. Delete events from GMT time 11:00AM, July 10, 2003 to GMT time 8:00PM, July 11, 2003:

   ```
   DeleteEvents –from 07/10/2003 11:00:00.000 –to
   07/11/2003 20:00:00.000 –z GMT
   ```

4. Delete events collected before 10 minutes together with statistic data from topology table:

    ```
    DeleteEvents –older 1 hr -minage 1 –statistics t
    ```

EventQueueCleanup

Location:

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

Purpose:

Cleans up messages on an event queue which are not being collected.

The utility optionally can remove all messages on the event queue which do not appear to be TransactionVision event messages. It also has the ability to remove or replay event messages which are targeted for a different database or were intended for a schema and/or communication link which no longer exists. These messages are replayed to an existing database, schema and communication link.

Syntax:

```
EventQueueCleanup
    { -showids }
    {
        -q EVENTQUEUE -qm QUEUEMANAGER
        [-c CHANNEL -h HOST [-p PORT]]
        [-erroneous]
        { [-replay SCHEMAID COMMLINKID | -delete]
            [-otherdbs] }
        [-verbose (-v)]
}
```

Options:

Option	Description
-showids	Display all schema and communication link IDs from the database defined in Database.properties.
-q EVENTQUEUE	Event queue to process messages on

Option	Description
-qm QUEUEMANAGER	Queue manager with event queue
-c CHANNEL	Client connection channel
-h HOST	Host name for client connection
-p PORT	Listener port for client connection (default: 1414)
-erroneous	Removes erroneous messages (messages not recognized as TransactionVision events) from the specified queue.
-replay SCHEMAID COM-MLINKID	Removes and replays events targeted to invalid schema and/or communication link IDs to new, valid schema and communication link IDs.
-delete	Removes and deletes events targeted to invalid schema and/or communication link IDs.
-otherdbs	Process events intended for other databases and consider them to have invalid schema and communication link IDs.
-verbose (-v)	Print verbose output.

Examples:

1. Show database IDs:

   ```
   EventQueueCleanup -showids
   ```

2. Delete all non-TransactionVision events from the event queue:

   ```
   EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE
   -erroneous
   ```

3. Delete all non-TransactionVision events and all events which are targeted to old, non-existent schemas or communication links:

   ```
   EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE
   -erroneous -delete
   ```

4. Replay all events which are targeted to old, non-existent schemas or communication links to a new schema and communication link so they can be processed by the Analyzer:

```
EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE
-replay 2 11
```

5. Show a snapshot of the queue without processing any messages (See Note 4):

```
EventQueueCleanup -qm QMGR1 -q TVISION.EVENT.QUEUE
-verbose
```

Notes:

1. Database Ids

Events on an event queue are targeted to a specific database, schema, and communication link. The schema and communication link are uniquely identified in the database by unique IDs. These IDs can be queried from the TVISION.SCHEMA and TVISION.PROJECT_COMMLINK or using the `EventQueueCleanup -showids` command.

When a schema or communication link is deleted (or changed in the case of communication links), the unique ID will no longer exist. Any events targeted for one of these old, deleted IDs still on the event queue, will never be processed.

This tool provides a means of either deleting these old, invalid events or replaying them for a new schema and communication link.

2. Erroneous Messages

An erroneous messages is any message on the event queue which is not recognized as a TransactionVision event message. This could include messages generated from other applications or WebSphere MQ COA report messages (Communication Link Time Skew Tests), which the analyzer may have timed out on because of WebSphere MQ infrastructure problems.

3. Other Databases

By default **EventQueueCleanup** will leave all event messages targeted to a different database. The `-otherdbs` option tells **EventQueueCleanup** to assume they are invalid and makes them eligible for deletion or replay based on the given option.

4. Verbose Output

The -verbose option can be used during regular processing (in conjunction with -erroneous, -delete, and/or -replay) to see what messages are left on the event queue, and which are deleted and/or replayed.

The -verbose option can also be used by itself to show a snapshot of the event queue without processing any of the messages on the queue.

FlushStatusUtil

Location:

TVISION_HOME/bin/FlushStatusUtil.[sh|bat]

Description:

Shows and cleans the flush status of cached statistics for the Static Component Topology Analysis view and analysis data. The status is stored in the database and determines whether recovery is needed when the Analyzer starts.

For cached statistics, the status contains the IDs of the events for which statistics and analysis data are missing in the database.

This utility obtains database connection information from the TransactionVision Database.properties file.

Syntax:

FlushStatusUtil SCHEMA [-dbcache | -stats] [-show |-clean]

Options:

Option	Description
schema	The name of the database schema to show or clean the flush status for.
-dbcache	Shows or cleans the current flush status of the analysis data cache.
-stats	Shows or cleans the current flush status of the cached statistics for the Static Component Topology Analysis view.

Option	Description
-show	Show the current flush status of the cached analysis data or statistics.
-clean	Clean the current flush status of the analysis data or statistics. The Analyzer will start up in a "clean" state without recovery, even though data might be missing in the database. Use this option with extreme caution.

Example Output:

The following output indicates that IDs 7916-8084 and the set of additional IDs are missing statistics and analysis data and need to be recovered at the next startup time:

```
Last written sequential_id in EVENT_LOOKUP: 8084
Last flushed sequential_id: 7915
Additional ids not flushed: 7912,7911,7901,7892,7870,7872
```

The following output indicates that all statistics and analysis data has been flushed; no recovery is needed:

```
Last written sequential_id in EVENT_LOOKUP: 9640
Last flushed sequential_id: 9640
```

GetPut

Location:

```
com.bristol.tvision.admin.GetPut
```

Description:

Transfers messages from the specified queue (typically the dead letter queue, but can be any queue) to the specified event queue to enable the Analyzer to attempt to reprocess the failed events.

Syntax:

```
GetPut -gq <GetQueueName> -pg <PutQueueName>
-qm <QueueManager> [-c <channel> -h <host> [-p <port>]]
```

OracleRunStats

Location:

TVISION_HOME/bin/OracleRunStats.[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 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 **cron** 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.

Option	Description
passwd	The password associated with *user_name*
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

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 |-exit [-quiesce][-project (-proj) PROJECTNAME]
  -status [-project (-proj) PROJECTNAME]
  -killserver
  -logfile
  -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.

Option	Description
`-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.
`-status`	Reports the current Analyzer status.
`-logconfig`	Reloads Analyzer logging configuration settings.
`-versioninfo`	Returns Analyzer version information.
`-killserver`	Shuts down the Analyzer immediately. This flag is not recommended; -exit is the preferred method for shutting down the Analyzer cleanly.
`-project (-proj) PRO-JECTNAME`	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, `-proj "Project Name"`). The project must have a communication link in order for the Analyzer to process events. Only the -start, -stop, and -status commands be performed on a single project. This option cannot be used in combination with the -host or -rmiregp options. When the -project option is used, the Analyzer host and port is looked up from the database.
`-host HOST`	Name of the host where the Analyzer runs. Can be either name or IP address. Local host is used if not specified.

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

Technical References

IBM OS/390 or z/OS MVS Initialization and Tuning Reference

MVS Initialization and Tuning Reference

IBM DB2 Administration Guide

WebSphere MQ Application Programming Guide (SC33-0807-08)

Managing WebLogic Security

Sensor Messages

Sensors log error messages in the system log of the computer where they are installed. UNIX system log, the Windows event log, the OS/390 operator console log, or the OS/400 user job log. In the messages below, %s and %1-n represent text strings that are filled in with appropriate information at runtime.

```
SLMS101E: TransactionVision Sensor: failed to allocate space for %s
```
Out of memory error. Contact Bristol Support.

```
SLMS102E: TransactionVision Sensor: failed to reallocate space for %s
```
Out of memory error. Contact Bristol Support.

```
SLMS103E: TransactionVision Sensor: failed to create mutex lock
```
Contact Bristol Support.

```
SLMS104E: TransactionVision Sensor: failed to lock mutex
```
Contact Bristol Support.

```
SLMS105E: TransactionVision Sensor: failed to unlock mutex
```
Contact Bristol Support.

```
SLMS106E: TransactionVision Sensor: failed to destroy mutex lock
```
Contact Bristol Support.

```
SLMS107E: TransactionVision Sensor: error processing %s
```
Error processing given field within a configuration message.

```
SLMS108E: TransactionVision Sensor: Cannot connect to configuration queue
          manager %s: %s
```
The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS109E: TransactionVision Sensor: Cannot open configuration queue %s on queue manager %s: %s

The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS10AE: TransactionVision Sensor: Cannot reconnect to queue manager %s after user's MQDISC call: %s

The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS10BE: TransactionVision Sensor: Cannot reopen configuration queue %s on queue manager %s after user's MQDISC call: %s

The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS10CE: TransactionVision Sensor: Cannot connect event queue manager %s: %s

The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS10DE: TransactionVision Sensor: Cannot open event queue %s on queue manager %s: %s

The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS10EE: TransactionVision Sensor: %d unsuccessful attempts were made to send an event message to queue ('%s' '%s'): %s Continuing to retry every %d ms...

Sensor was unable to put an event message to the given event queue. Often, this is a result that occurs if the event queue becomes full if the Sensor is producing event messages faster than the Analyzer can pull them off the queue. The Sensor will continue to try to put to the event queue while the Analyzer is catches up. For information about configuring retry attempts, see page 172.

SLMS10FE: TransactionVision Sensor: Timed out (%d s) attempting %u times to put event message to queue ('%s' '%s')"

Sensor was unable to put an event message to the given queue after the stated number of attempts. Often the is a result that occurs if the event queue is full. For information about configuring retry attempts, see page 172.

SLMS110I: `TransactionVision Sensor: successfully put an event message on queue ('%s' '%s') after %d unsuccessful attempts`
Informational message stating that the Sensor has successfully put an event message on the event queue after it had to retry multiple times. For information about configuring retry attempts, see page 172.

SLMS111E: `TransactionVision Sensor: tried to use invalid handle for putting to event queue ('%s' '%s')`
An invalid handle was used to try to put to the event queue. This can occur if an application corrupts or passes a bad Hconn value to subsequent WebSphere MQ calls. The result is no event is sent to the Analyzer.

SLMS112E: `TransactionVision Sensor: Event message too big for queue ('%s' '%s') max message length. This error is reported only once per event queue and could be occurring more than once. Please increase max message length setting on the event queue.`
The event queue max message length size is too small for an event message. For information about setting maximum message length, see "Edit a Communication Link or Template" on page 176.

SLMS113W: `TransactionVision Sensor: attempt to disable crossing exit sensor failed"`
An internal error has occurred. Contact Bristol Support.

SLMS114W: `TransactionVision Sensor: crossing exit sensor has been disabled since both are active"`
If the standard TransactionVision WebSphere MQ Sensor library is active, the Crossing Exit Sensor is automatically disabled. Only one type of WebSphere MQ Sensor can be active at any time.

SLMS115W: `TransactionVision Sensor: unexpected internal error - Sensor turned off.`
An internal error has occurred. The WebSphere MQ application will continue to run, but events will no longer be reported. Contact Bristol Support.

SLMS116W: `TransactionVision Sensor: failed to close %s queue '%s' '%s'`
The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS117W: `TransactionVision Sensor: failed to get %s`
The specified WebSphere MQ error occurred. See WebSphere MQ documentation for explanation.

SLMS118W: TransactionVision Sensor: error removing configuration message
from queue ('%s' '%s'): %s
Failed to destructively get an expired or invalid configuration message from the
configuration queue because of the specified WebSphere MQ error.

SLMS119W: TransactionVision Sensor: cannot extract %s from configuration
message
Failed to extract a given field from a configuration message. Possibly an invalid
configuration message.

SLMS11AE: TransactionVision Sensor: error in communication infrastructure
%s: %s
An error occurred with the TransactionVision communication infrastructure. A
WebSphere MQ error is included with the error.

SLMS11BI: TransactionVision Sensor: errors on queue manager ('%s') will be
supressed until successful communication resumes"
Informational message indicating that the Sensor will discontinue logging errors on the
specified queue manager until it is able to communicate with the queue manager once
again.

SLMS11CI: TransactionVision Sensor: Successful communication has been made
to the queue manager ('%s'). Logging will resume."
Informational trace message indicating that the Sensor is able to communication with
the specified queue manager. Error logging had been supressed for the queue manager
while the Sensor was unable to communicate with it, but will now resume.

SLMS11DI: TransactionVision Sensor: errors on configuration queue ('%s'
'%s') will be supressed until successful communication resumes"
Informational message indicating that the Sensor will discontinue logging errors on the
specified configuration queue until it is able to communicate with the queue once
again.

SLMS11EI: TransactionVision Sensor: Successful communication has been made
to the configuration queue ('%s' '%s'). Logging will resume."
Informational trace message indicating that the Sensor is able to communication with
the specified configuration queue. Error logging had been supressed for the queue
while the Sensor was unable to communicate with it, but will now resume.

SLMS11FW: TransactionVision Sensor: failed to unmarshall configuration
message
The Sensor failed to unmarshall a configuration message. Possibly an invalid
configuration message.

SLMS120W: TransactionVision Sensor: invalid %s version

An invalid version of a given WebSphere MQ structure has been encountered when constructing the event message.

SLMS121W: TransactionVision Sensor: invalid message on configuration queue ('%s' '%s') (not minimum configuration message size) (putting to dead letter queue)

An invalid configuration message has been encountered on the configuration queue. The Sensor will put the invalid message on the dead letter queue.

SLMS122W:TransactionVision Sensor: invalid message on configuration queue ('%s' '%s') (configuration length does not match message length) (putting to dead letter queue)"

An invalid configuration message has been encountered on the configuration queue. The Sensor will put the invalid message on the dead letter queue.

SLMS123W:TransactionVision Sensor: invalid message on configuration queue ('%s' '%s') (product name incorrect in configuration message) (putting to dead letter queue)"

An invalid configuration message has been encountered on the configuration queue. The Sensor will put the invalid message on the dead letter queue.

SLMS124W: TransactionVision Sensor: unsupported configuration message version found on ('%s' '%s') (putting to dead letter queue)

An invalid configuration message has been encountered on the configuration queue. The Sensor will put the invalid message on the dead letter queue.

SLMS125W:TransactionVision Sensor: failed to remove configuration message from configuration queue ('%s' '%s') and put to dead letter queue ('%s' '%s'): %s"

The Sensor was unable to remove a configuration message from the configuration queue and place it on the dead letter queue. A WebSphere MQ error accompanies this message.

SLMS126W: TransactionVision Sensor: invalid data range value in configuration message

Contact Bristol Support.

SLMS127W: TransactionVision Sensor: missing expected token separator in configuration message when parsing %s

Contact Bristol Support.

SLMS128W: TransactionVision Sensor: missing expected space separator in
           configuration message when parsing %s
Contact Bristol Support.

SLMS129W: TransactionVision Sensor: missing expected subsection separator
           in configuration message when parsing %s
Contact Bristol Support.

SLMS12AW: TransactionVision Sensor: missing expected section separator in
           configuration message when parsing %s
Contact Bristol Support.

SLMS12BW: TransactionVision Sensor: missing expected data range dash in
           configuration message
Contact Bristol Support.

SLMS12CE: TransactionVision Sensor: bad checksum for %s
Contact Bristol Support.

SLMS12DI: TransactionVision Sensor: opening configuration queue '%s' '%s'
Informational trace message.

SLMS12EI: TransactionVision Sensor: opening event queue '%s' '%s'
Informational trace message.

SLMS12FI: TransactionVision Sensor: got configuration message from '%s'
           '%s'
Informational trace message.

SLMS130I: TransactionVision Sensor: invalidating all previous configuration
           messages on '%s' '%s' from source analyzer
Informational trace message. Invalidating and removing all configurations from a
given Analyzer previous to the configuration message currently being processed.

SLMS131I: TransactionVision Sensor: invalidating specified configuration
           message on '%s' '%s' from source analyzer
Informational trace message. Invalidating and removing a given configuration message
from a given Analyzer specified by the configuration message currently being
processed.

SLMS132I: TransactionVision Sensor: removing invalidated configuration
           message from '%s' '%s' MsgId: %s CorrelId: %s
Informational trace message.

```
SLMS133I: TransactionVision Sensor: removing expired configuration message
          from '%s' '%s' current GMT: %s expire timE:%s MsgId: %s
          CorrelId: %s
```
Informational trace message.

```
SLMS134I: TransactionVision Sensor: already have this configuration message
          from '%s' '%s'
```
Informational trace message. A duplicate configuration message has been found on the configuration queue.

```
SLMS135I: TransactionVision Sensor: event message (%u bytes) too large for
          event queue. Removing user data to make event message smaller.
```
The event will be included in the project, but the user data buffer will not be available.

```
SLMS136I: TransactionVision Sensor: this host ('%s') does not appear in
          filter rule
```
The host the Sensor is running on does not match the collection filter criteria for hosts.

```
SLMS137I: TransactionVision Sensor: this user name ('%s') does not appear
          in filter rule
```
The user name associated with the message does not match the collection filter criteria for user names.

```
SLMS138I: TransactionVision Sensor: this program ('%s') does not appear in
          filter rule
```
The program name of the program using the Sensor does not match the collection filter criteria.

```
SLMS139I: TransactionVision Sensor: current time is out of filter
          start/stop time rangE:current GMT: %s start timE:%s stop timE:%s
```
The current time does not match the collection time specified in the collection filter.

```
SLMS13AI: TransactionVision Sensor: this region id ('%s') does not appear
          in filter rule
```
The region id does not match the collection filter criteria.

```
SLMS13BI: TransactionVision Sensor: this transaction id ('%s') does not
          appear in filter rule
```
The transaction id does not match the collection filter criteria.

```
SLMS13CI: TransactionVision Sensor: this job name ('%s') and step ('%s')
          pair does not appear in filter rule
```
The OS/390 Batch job name and step combination does not match the collection filter criteria.

```
SLMS13DI: TransactionVision Sensor: the job name ('%s') does not appear in
          filter rule
```
The OS/400 job name does not match the collection filter criteria.

```
SLMS13EI: TransactionVision Sensor: the IMS identifier ('%s') does not
          appear in filter rule
```
The IMS identifier does not match the collection filter criteria.

```
SLMS13FI: TransactionVision Sensor: the IMS region type ('%s') does not
          appear in filter rule
```
The IMS region type does not match the collection filter criteria.

```
SLMS140I: TransactionVision Sensor: the IMS region identifier ('%s') does
          not appear in filter rule
```
The IMS region identifier does not match the collection filtering criteria.

```
SLMS141I: TransactionVision Sensor: the IMS transaction name ('%s') does
          not appear in filter rule
```
The IMS transaction name does not match the collection filtering criteria.

```
SLMS142I: TransactionVision Sensor: the IMS PSB name ('%s') does not appear
          in filter rule
```
The IMS PSB name does not match the collection filtering criteria.

```
SLMS143I: TransactionVision Sensor: API name ('%s') does not appear in
          filter rule
```
The WebSphere MQ API does not match the collection filter criteria.

```
SLMS144I: TransactionVision Sensor: completion code (%u) does not appear in
          filter rule
```
The WebSphere MQ API completion code does not match the collection filter criteria.

```
SLMS145I: TransactionVision Sensor: reason code (%u) does not appear in
          filter rule
```
The WebSphere MQ API reason code does not match the collection filter criteria.

```
SLMS146I: TransactionVision Sensor: technology ('%s') in configuration does
          not match this technology ('%s')
```
The event technology does not match the collection filter criteria.

```
SLMS147I: TransactionVision Sensor: queue manager does not appear in filter
          rulE:'%s'"
```
The queue manager does not match the data collection filter criteria.

SLMS148I: TransactionVision Sensor: queue manager and/or object name does
          not appear in filter rule
          The queue manager or WebSphere MQ object name does match the collection filter
          criteria.

SLMS149I: TransactionVision Sensor: ReplyTo queue manager and/or queue name
          does not appear in filter rule: '%s' '%s'"
          The ReplyTo queue manager or queue name does not match the collection filter
          criteria.

SLMS14AI: TransactionVision Sensor: severity (%d) does not appear in filter
          rule
          The BTTRACE severity does not match the collection filter criteria.

SLMS14BI: TransactionVision Sensor: collection mask does not match %s for
          %s
          The specified entity does not match the collection filter criteria.

SLMS14CI: TransactionVision Sensor: this MQSI broker name ('%s') does not
          appear in filter rule
          The MQSI broker name does not match the collection filter criteria.

SLMS14DI: TransactionVision Sensor: this MQSI message flow name ('%s') does
          not appear in filter rule
          The MQSI message flow name does not match the collection filter criteria.

SLMS14EI: TransactionVision Sensor: MQCMIT has been filtered out because
          all other events in the unit of work have been filtered out"
          The MQCMIT was filtered out as specified in the collection filter criteria.

SLMS14FI: TransactionVision Sensor: Exit part of MQDISC has been filtered
          out"
          The MQDISC exit was filtered out as specified in the collection filter criteria.

SLMS150I: TransactionVision Sensor: ignoring unmatched configuration
          message from '%s' '%s'
          Ignoring a configuration message because a standard collection filter criteria did not
          match with the running Sensor.

SLMS151I: TransactionVision Sensor: creating filter rule from matched
          configuration message from '%s' '%s' expire timE:%s MsgID: %s
          CorrelId: %s
          The Sensor matches the standard collection filter criteria of a given configuration
          message.

SLMS152I: TransactionVision Sensor: removing expired filter rule from
        memory current GMT: %s expire timE:%s
        A configuration message that has already been retrieved from the configuration queue
        has expired.

SLMS153I: putting event message on event queue ('%s' '%s')
        Informational trace message.

SLMS154I: Management Initiated for MQSeries API Crossing Exit
        Informational trace message.

SLMS155I: TransactionVision Sensor: Disabling MQSeries API Crossing Exit
        Informational trace message.

SLMS156I: TransactionVision Sensor: Enabling MQSeries API Crossing Exit
        Informational trace message.

SLMS157I: TransactionVision Sensor: SLMC Exiting
        Informational trace message.

SLMS158I: TransactionVision Sensor: The Sensor is being used with another
        TransactionVision component. Please remove the Sensor path from
        the library search path before starting the analyzer or Java
        sensor."
        The system library path environment variable has been set to load the Sensor instead of
        the real WebSphere MQ library. This library path should be set only for applications
        being monitored, but not for the Analyzer or the Servlet, JMS, or EJB Sensors.

SLMS159E: TransactionVision Sensor: NULL parameter passed to API exit"
        An internal error has occurred. Contact Bristol Support.

SLMS15AE: TransactionVision Sensor: failed to register API exit: %d"
        An internal error has occurred. Contact Bristol Support.

SLMS15BE: TransactionVision Sensor: failed to allocate space for %s,
        package size=%d,package count=%d,max size=%d."
        The Sensor was unable to alloacte sufficient space for the event package because the
        package size exceeds the maximum size.

SLMS202S: TransactionVision Sensor: cannot initialize MQSeries library
        A general error reporting that the loading and initialization of the real WebSphere MQ
        library failed. WebSphere MQ may not be properly installed, or the library may not be
        in the library search path.

SLMS252E: TransactionVision Sensor: could not connect to queue manager:
'%s'"

The Sensor was unable to connect to the queue manager (a WebSphere MQ message is supplied). For information about setting connection retry attempts, see "Edit a Communication Link or Template" on page 176.

SLMS253E: TransactionVision Sensor: could not connect to queue manager
'%s': %s The Sensor will attempt to reconnect after %d seconds.
Events may not be collected during this time."

The Sensor was unable to connect to the queue manager (a WebSphere MQ message is supplied). It will attempt to reconnect, but no events will be collected until it succeeds. For information about setting connection retry attempts, see "Edit a Communication Link or Template" on page 176.

SLMS254E: TransactionVision Sensor: Error marshalling event data: null
pointer '%s'"

Contact Bristol Support.

SLMS255E: TransactionVision Sensor: Unknown error occurred (type=%d), Event
Dispatcher is shutting down."

Contact Bristol Support.

SLMS256I: TransactionVision Sensor: The routine that generated abend in
Event Dispatcher is '%.80s', and offset is %08X. Please contact
Bristol Support with dump information."

Contact Bristol Support.

SLMS257E: TransactionVision Sensor: Wrong number of parameters. Please
specify Queue Manager name only."

Too many parameters were specified; specify the Queue Manager name only.

SLMS258E: TransactionVision Sensor: Input queue manager name (%s) length is
bigger than %d (MQ limit)."

The queue manager name is too long.

SLMS259E: TransactionVision Sensor: Event Dispatcher initialization
failed."

Contact Bristol Support.

SLMS25AI: TransactionVision Sensor: Event Dispatcher initialization
completed."

Informational trace message.

```
SLMS25BI: TransactionVision Sensor: Event Dispatcher is ending as requested
          by TVISIONB."
```
Informational trace message.

```
SLMS302I: TransactionVision WMQI Sensor: %s
```
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

```
SLMS303I: TransactionVision WMQI Sensor: inserting attribute '%s'
```
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

```
SLMS304I: TransactionVision WMQI Sensor: inserting %s terminal entry for
          terminal '%s'
```
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

```
SLMS305I: TransactionVision WMQI Sensor: getting attribute by index %d
```
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

```
SLMS306I: TransactionVision WMQI Sensor: getting attribute '%s'
```
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

```
SLMS307I: TransactionVision WMQI Sensor: setting attribute '%s' to '%s'
```
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

```
SLMS308E: TransactionVision WMQI Sensor: failed to allocate space for %s
```
Out of memory error. Contact Bristol Support.

```
SLMS309E: TransactionVision WMQI Sensor: failed to initialize %s
```
Contact Bristol Support.

```
SLMS30AE: TransactionVision WMQI Sensor: invalid data type for %s
```
Contact Bristol Support.

```
SLMS30BE: TransactionVision WMQI Sensor: failure terminal does not exist
```
Contact Bristol Support.

```
SLMS30CE: TransactionVision WMQI Sensor: out terminal does not exist
```
Contact Bristol Support.

```
SLMS30DE: TransactionVision WMQI Sensor: out terminal is not attached
```
Contact Bristol Support.

```
SLMS30EE: TransactionVision WMQI Sensor: unknown attribute '%s'
```
Contact Bristol Support.

SLMS30FE: `TransactionVision WMQI Sensor: attribute index %d not found`
Contact Bristol Support.

SLMS310E: `TransactionVision WMQI Sensor: %s terminal '%s' not found`
Contact Bristol Support.

SLMS311I: `TransactionVision WMQI Sensor: loading TransactionVision Sensor`
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

SLMS312E: `TransactionVision WMQI Sensor: failed to load TransactionVision`
`Sensor`
Ensure that the Sensor is properly installed. See the *TransactionVision Installation and Configuration Guide* for installation instructions.

SLMS313I: `TransactionVision WMQI Sensor: TransactionVision Sensor`
`successfully loaded`
Informational message regarding the WebSphere MQ Integrator-enabled Sensor.

SLMS314E: `TransactionVision WMQI Sensor: Could not find Path value in the`
`registry. Ensure that the TransactionVision Sensor is properly`
`installed.`
Ensure that the Sensor is properly installed. See the *TransactionVision Installation and Configuration Guide* for installation and configuration instructions.

SLMS315E: `TransactionVision WMQI Sensor: Could not open TransactionVision`
`Sensor key in registry. Ensure that the TransactionVision Sensor`
`is properly installed.`
Ensure that the Sensor is properly installed. See the *TransactionVision Installation and Configuration Guide* for installation and configuration instructions.

SLMS316E: `TransactionVision WMQI Sensor: could not load TransactionVision`
`Sensor library ('%s'): %s`
Ensure that the Sensor is properly installed. See the *TransactionVision Installation and Configuration Guide* for installation and configuration instructions.

SLMS400I: `TVISIONB startup in progress.`
Informational message. TVISIONB has been started and startup is in progress. This message should be followed by SLMS401I when startup is complete.

SLMS401I: `TVISIONB startup complete.`
Informational message. TVISIONB startup is complete and the MQSeries-IMS bridge Sensor server component is ready to receive events from the bridge.

SLMS402I: STOP command received.

Informational message. The MQSeries-IMS bridge Sensor server component is shutting down. Bridge monitoring is disabled to prevent the sending of further events from the OTMA Input/Output Edit exit routine in the IMS control region(s) and all tasks in the TVISIONB address space are terminated. This message should be followed by SLMS403I when termination is complete. There may be a delay while in-flight events are dispatched.

SLMS403I: TVISIONB termination complete.

Informational message. All tasks in the MQSeries-IMS bridge Sensor server component have been terminated. The TVISIONB and TVISIOND address spaces have been terminated.

SLMS4041: TVISIONB termination in progress.

TVISIONB is terminating in response to a STOP command or due to an error condition.

SLMS4051: TVISIONB dispatching remaining events in buffer queue.

TVISIONB is in quiesce mode in response to a STOP command. When all events in the buffer queue have been dispatched, TVISIONB termination will complete. The SLMS415I message follows to report the number of events in the buffer queue.

SLMS4061: Immediate TVISIONB termination requested. Events in buffer queue will be lost.

TVISIONB is terminating in response to a STOP IMMED command issued while there are events in the buffer queue. The SLMS415I message follows to report the number of events in the buffer queue.

SLMS410I: TVISIONB MQIMSBDG monitoring disabled.

Response to a DISABLE MQIMSBDG command. MQSeries-IMS bridge monitoring is disabled regardless of Analyzer requests.

SLMS411W: TVISIONB configuration messages have requested MQIMSBDG monitoring, which has been disabled by the system operator. Monitoring will not resume until enabled by the operator.

Warning message issued when configuration messages from an Analyzer request MQSeries-IMS bridge monitoring but monitoring has been disabled by a DISABLE MQIMSBDG command.

SLMS412I: TVISIONB MQIMSBDG monitoring already disabled.
Response to a DISABLE MQIMSBDG command when MQSeries-IMS bridge
monitoring has already been disabled by a previous DISABLE MQIMSBDG
command.

SLMS413I: TVISIONB MQIMSBDG monitoring enabled.
Response to an ENABLE MQIMSBDG command. MQSeries-IMS bridge monitoring
is enabled.

SLMS414I: TVISIONB MQIMSBDG monitoring already enabled.
Response to an ENABLE MQIMSBDG command when MQSeries-IMS bridge
monitoring is already enabled.

SLMS419E: TVISIONB unable to process command.
An invalid command format or unknown command has been issued or TVISIONB
encountered a system error attempting to process a command. If the command is
correct, please contact Bristol Support and report the diagnostic data included in the
message.

SLMS420W: Possible TVISIONB stall condition detected. Some requested events
may be not be collected.
The TVISIONB event recording function is not responding in a timely manner. This
condition may be caused by a full TransactionVision event queue because the queue
has been defined too small or no Analyzer service is running to relieve the queue. If the
event queue is full, expand its size and/or assure that an Analyzer is running.
Otherwise, please contact Bristol Support.

SLMS421S: The event dispatcher task has unexpectedly terminated. TVISIONB
will terminate.
TVISIOND is terminating due to an error condition.

SLMS430E: TVISIONB already started.
An second instance of TVISIONB has been started. Only one instance is allowed. The
second instance will be terminated.

SLMS431E: The MQSeries queue manager name missing. Please specify as
follows: S TVISIONB,QMGR=CSQ1, for example.
The WebSphere MQ queue manager, to which TVISIONB connects to access its
configuration and event queues, must be identified at startup.

SLMS432E: The MQSeries queue manager name specified on the S TVISIONB
command contains more than the maximum of 4 characters.
WebSphere MQ queue manager names on OS/390 are limited to four characters.

SLMS433E: The MAXQ size specified on the S TVISIONB command is less than
          the minimum requirement of 3.
          At least 3MB of storage are required for TVISIONB to manage its buffer queue.

SLMS434E: The MAXQ size specified on the S TVISIONB command is greater than
          the maximum allowed of 2046.
          The maximum amount of storage that TVISIONB can use to manage its buffer queue is
          2046MB (2GB less 1MB).

SLMS435E: The MAXQ size specified on the S TVISIONB command contains non-
          numeric characters.
          The TVISIONB MAXQ parameter must be numeric.

SLMS436E: The MAXQ size specified on the S TVISIONB command contains too
          many characters.
          The TVISIONB MAXQ parameter must contain a maximum of 15 characters.

SLMS438E: The event dispatcher startup procedure name contains more than
          the maximum of 8 characters.
          The EDPROC parameter on the TVISIONB startup command or procedure is
          incorrect.

SLMS440E: Invalid TVISIONB command entered.
          An invalid command format or unknown command has been issued.

SLMS441E: Unknown target of TVISIONB DISABLE command.
          A DISABLE command has been issued but the object name specified is unknown. In
          this version of TransactionVision, the only supported object is MQIMSBDG.

SLMS442E: Unknown target of TVISIONB ENABLE command.
          A ENABLE command has been issued but the object name specified is unknown. In
          this version of TransactionVision, the only supported object is MQIMSBDG.

SLMS450S: TVISIONB ENQ error. Diagnostic data follow:
          Please contact Bristol Support and report the diagnostic data included in the message.

SLMS451S: TVISIONB unable to allocate required storage. Diagnostic data
          follow:
          This may be caused by an too small REGION size for TVISIONB, which requires an
          address space size of the MAXQ specification plus 1MB. If the REGION size is
          sufficient, please contact Bristol Support and report the diagnostic data included in the
          message.

SLMS452S: TVISIONB unable to release obtained storage. Diagnostic data
          follow:
Please contact Bristol Support and report the diagnostic data included in the message.

SLMS453S: TVISIONB ATTACH error. Diagnostic data follow:
This may be caused by a required program module missing from STEPLIB or
SSLMLOAD libraries specified in the TVISIONB startup procedure. Please assure that
all the modules listed in the installation instructions are located in the appropriate
libraries and those libraries are specified in the TVISIONB startup procedure. If the
setup is correct, please contact Bristol Support and report the diagnostic data included
in the message.

SLMS454S: TVISIONB LINK error. Diagnostic data follow:
This may be caused by a required program module missing from STEPLIB or
SSLMLOAD libraries specified in the TVISIONB startup procedure. Please assure that
all the modules listed in the installation instructions are located in the appropriate
libraries and those libraries are specified in the TVISIONB startup procedure. If the
setup is correct, please contact Bristol Support and report the diagnostic data included
in the message.

SLMS455S: TVISIONB LOAD error. Diagnostic data follow:
This may be caused by a required program module missing from STEPLIB or
SSLMLOAD libraries specified in the TVISIONB startup procedure. Please assure that
all the modules listed in the installation instructions are located in the appropriate
libraries and those libraries are specified in the TVISIONB startup procedure. If the
setup is correct, please contact Bristol Support and report the diagnostic data included
in the message.

SLMS456S: TVISIONB DELETE error. Diagnostic data follow:
Please contact Bristol Support and report the diagnostic data included in the message.

SLMS457S: TVISIONB OPEN error. Diagnostic data follow:
Please contact Bristol Support and report the diagnostic data included in the message.

SLMS458S: TVISIONB CLOSE error. Diagnostic data follow:
Please contact Bristol Support and report the diagnostic data included in the message.

SLMS459S: TVISIONB BLDL error. Diagnostic data follow:
Please contact Bristol Support and report the diagnostic data included in the message.

SLMS460S: TVISIONB ASCE error. Diagnostic data follow:
Please contact Bristol Support and report the diagnostic data included in the message.

```
SLMS461S: TVISIONB ASEXT error. Diagnostic data follow:
```
Please contact Bristol Support and report the diagnostic data included in the message.

```
SLMS472S: TVISIONB IEANTCR error. Diagnostic data follow:
```
Please contact Bristol Support and report the diagnostic data included in the message.

```
SLMS473S: TVISIONB IEANTRT error. Diagnostic data follow:
```
Please contact Bristol Support and report the diagnostic data included in the message.

```
SLMS474S: TVISIONB IEANTDL error. Diagnostic data follow:
```
Please contact Bristol Support and report the diagnostic data included in the message.

```
SLMS475S: TVISIONB ESTAEX error. Diagnostic data follow:
```
Please contact Bristol Support and report the diagnostic data included in the message.

```
SLMS485S: Insufficient storage available for TVISIONB queue requirements.
          Diagnostic data follow:
```
This may be caused by a too small REGION size for TVISIONB, which requires an address space size of the MAXQ specification plus 1MB. This error occurs when TVISIONB attempts to allocate its initial buffer queue space of 2MB during startup. TVISIONB is terminated. If the REGION size is sufficient, please contact Bristol Support and report the diagnostic data included in the message.

```
SLMS486W: Insufficient storage available for TVISIONB queue expansion
          requirements. Events may be lost.
```
This may be caused by a too small REGION size for TVISIONB, which requires an address space size of the MAXQ specification plus 1MB. This error occurs when TVISIONB attempts to allocate 1MB to expand its buffer queue space within the MAXQ limitation specified at TVISIONB startup. When all already-allocated queue space is exhausted, subsequent events will be discarded. This condition may be caused by a full TransactionVision event queue because the queue has been defined too small or no Analyzer service is running to relieve the queue. If the event queue is full, expand its size and/or assure that an Analyzer is running. If the event queue is not full and the REGION size is sufficient, please contact Bristol Support.

SLMS487W: Insufficient storage available for TVISIONB queue expansion
requirements. MAXQ limit reached. Events may be lost.
TVISIONB has used almost all of the buffer queue space allowed by the MAXQ
limitation specified at TVISIONB startup. When all allowed queue space is exhausted,
subsequent events will be discarded. This condition may be caused by a full
TransactionVision event queue because the queue has been defined too small or no
Analyzer service is running to relieve the queue. If the event queue is full, expand its
size and/or assure that an Analyzer is running. Otherwise, increase the MAXQ
specification.

SLMS488S: TVISIONB system stall condition detected. Queued events will be
lost.
Please contact Bristol Support and report the diagnostic data included in the message.

SLMS498S: TVISIONB system error. Diagnostic data follow:
Please contact Bristol Support and report the diagnostic data included in the message.

SLMS499S: TVISIONB system error. Diagnostic data follow:
Please contact Bristol Support and report the diagnostic data included in the message.

SLDS272E: TransactionVision Sensor: wrong number of parameters."
The Sensor command contains an incorrect number of parameters. See Chapter 9,
"Configuring the CICS Sensor," in the *TransactionVision Installation and
Configuration Guide* for instructions on starting and stopping the CICS Sensor.

SLDS273E: TransactionVision Sensor: input queue manager name (%s) length is
bigger than %d (MQ limit)."
The specified queue manager name exceeds the allowed length.

SLDS274E: FailMemCDC: E: "%s%03XE %s:TransactionVision Sensor: Failed to
allocate memory for SLDI."
Out of memory error. The CICS Sensor will fail to initialize and stop. Please try to
restart CICS Sensor, if the same problem happens, contact Bristol support.

SLDS275E: FailMemCCT: E: "%s%03XE %s:TransactionVision Sensor: Failed to
allocate memory for config table."
Out of memory error. The CICS Sensor will fail to initialize and stop. Please try to
restart CICS Sensor, if the same problem happens, contact Bristol support.

SLDS276E: TransactionVision Sensor: Unknown error occurred (type=%d), CICS
Sensor Driver is shutting down."
Please contact Bristol Support.

```
SLDS277I: TransactionVision Sensor: The routine that generated abend in
          CICS Sensor Driver is '%.80s', and offset is %08X. Please contact
          Bristol Support with the dump information."
```
Please contact Bristol Support and report the diagnostic information included in the message.

```
SLDS278E: InitFailed: E: "%s%03XE %s: TransactionVision Sensor: CICS %s
          Sensor Driver startup failed. QMGR=%s, CONFIGQ=%s."
```
CICS Sensor driver program failed to initialize due to error(s) previous shown.

```
SLDS279I: TransactionVision Sensor: CICS %s Sensor Driver startup
          completed. QMGR=%s, CONFIGQ=%s."
```
Informational message indicating the CICS Sensor Driver startup completed successfully.

```
SLDS27AE: WrongEventType: E: "%s%03XE %s: TransactionVision Sensor: Unknown
          event type (type=%d), skip this event."
```
The event type is not collectable by the CICS Sensor, and the event is skipped.

```
SLDS27BI: TransactionVision Sensor: CICS %s Sensor Driver is ending, return
          code = %d."
```
Informational message indicating the CICS Sensor Driver is ending.

```
SLDS27CI: TransactionVision Sensor: this terminal id ('%s') does not appear
          in filter rule"
```
The terminal ID does not match the collection filter criteria.

```
SLDS27DI: TransactionVision Sensor: this API type ('%d') does not appear in
          filter rule"
```
The API type does not match the collection filter criteria.

```
SLDS27EI: TransactionVision Sensor: this API ('%d') does not appear in
          filter rule"
```
The API does not match the collection filter criteria.

```
SLDS27FI: TransactionVision Sensor: this file ('%s') does not appear in
          filter rule"
```
The file does not match the collection filter criteria.

```
SLDS280I: TransactionVision Sensor: this TD queue ('%s') does not appear in
          filter rule"
```
The TD queue does not match the collection filter criteria.

SLDS281I: `TransactionVision Sensor: this TS queue ('%s') does not appear in filter rule"`

The TS queue does not match the collection filter criteria.

SLDS282I: `TransactionVision Sensor: this EIBRESP ('%d') does not appear in filter rule"`

The response does not match the collection filter criteria.

SLDS400I: `TVISION [TransactionVision Manager | CICS sysid TransactionVision sensor] startup in progress.`

Informational message issued in response to a start TransactionVision Manager command or a start Sensor command, where *sysid* is the SYSID of the CICS region to monitor.

SLDS401I: `TVISION [TransactionVision Manager tvid | CICS sysid TransactionVision sensor] startup complete.`

Informational message issued when TransactionVision Manager startup or Sensor startup is complete, where *tvid* is the TVID of the TransactionVision Manager and *sysid* is the SYSID of the CICS region to monitor.

SLDS402I: `tvid STOP command received.`

Informational message issued in response to a stop TransactionVision Manager command, where *tvid* is the TVID of the TransactionVision Manager.

SLDS404I: `TVISION [TransactionVision Manager tvid | CICS sysid TransactionVision sensor] termination in progress.`

IInformational message issued in response to a stop TransactionVision Manager command or a stop Sensor command, where *tvid* is the TVID of the TransactionVision Manager and *sysid* is the SYSID of the monitored CICS region.

SLDS405I: `TVISION [TransactionVision Manager tvid | CICS sysid TransactionVision sensor] termination complete.`

Informational message issued when TransactionVision Manager shutdown or Sensor shutdown is complete, where *tvid* is the TVID of the TransactionVision Manager and *sysid* is the SYSID of the monitored CICS region.

SLDS406E: `No parameters specified on the START command. The TVID parameter is required.`

A start TransactionVision Manager command was issued without the required TVID parameter.

SLDS407E: `The TVID parameter is missing from the START command.`

A start TransactionVision Manager command was issued without the required TVID parameter.

SLDS408E: `The TVID specified on the START command contains more than the maximum of 4 characters.`

A start TransactionVision Manager command specified an invalid TVID parameter.

SLDS409E: `The SYSID parameter is missing from the START command.`

An incorrect START command has been issued and the TransactionVision CICS A start sensor command was issued without the required SYSID parameter.

SLDS410E: `The SYSID specified on the START command contains more than the maximum of 4 characters.`

A start sensor command specified an invalid SYSID parameter.

SLDS411E: `TVISION` *tvid* `TransactionVision Manager already started.`

A start TransactionVision Manager command specified a TVID value that matches a running TransactionVision Manager, where *tvid* is the TVID specified.

SLDS412E: `TVISION CICS` *sysid* `TransactionVision sensor already started.`

A start Sensor command specified a SYSID value that matches a running sensor under control of the same TransactionVision Manager, where *sysid* is the SYSID specified.

SLDS413E: `TVISION CICS` *sysid* `TransactionVision sensor already started by another TransactionVision Manager.`

A start Sensor command specified a SYSID value that matches a running sensor under control of a different TransactionVision Manager, where *sysid* is the SYSID specified.

SLDS415E: `TVISION The` *parameter* `value specified on the command is less than the required minimum of` *min_value*.

A start Sensor command specified a parameter value less than the minimum required, where *parameter* is the parameter keyword and *min_value* is the minimum value allowed.

SLDS416E: `TVISION The` *parameter* `value specified on the command is greater than the maximum allowed of` *max_value*.

A start Sensor command specified a parameter value more than the maximum required, where *parameter* is the parameter keyword and *max_value* is the maximum value allowed.

```
SLDS418E:  TVISION The parameter value specified on the command contains
           non-numeric characters.
```
A start Sensor command specified non-numeric characters for a parameter value that is required to be numeric, where *parameter* is the parameter keyword.

```
SLDS420E:  TVISION Command name, command_name, invalid.  Command ignored.
```
An invalid command was issued, where *command_name* is the command.

```
SLDS421E:  TVISION Command operand, operand_name, invalid for command_name
           command.  Command ignored.
```
An invalid command operand was specified, where *operand_name* is operand specified and *command_name* is the command.

```
SLDS422E:  TVISION  Operand value specified on command name or operand that
           takes no value specification.  Command ignored.
```
A *name(value)* form was specified on a command when no value was expected for the command or operand name specified.

```
SLDS423E:  TVISION Syntax error - misplaced ''(''.  Command ignored.
```
Syntax error.

```
SLDS424E:  TVISION Syntax error - misplaced '')''.  Command ignored.
```
Syntax error.

```
SLDS425E:  TVISION Command input is all spaces.  Command ignored.
```
Syntax error.

```
SLDS427E:  TVISION The sensor driver startup procedure name contains more
           than the maximum of 8 characters.
```
The value of the DRVRPROC parameter on a start Sensor command is invalid.

```
SLDS428E:  TVISION The required operand value for the keyword operand,
           operand_name, is missing.
```
A value was not specified for an operand that required a value, where *operand_name* is the operand name.

```
SLDS429E:  TVISION The required sensor type operand is missing.
```
The Sensor type, CICS, was omitted from the command. In this release, CICS is the only Sensor type supported but subsequent releases will add other types.

```
SLDS430E:  TVISION Invalid sensor type specified.
```
The only valid Sensor type in this release is CICS.

```
SLDS431E:  TVISION Unsupported sensor type specified.
```
The only valid Sensor type in this release is CICS.

SLDS432E: TVISION The required *command_name* command operand, *operand_name*,
is missing.
A required operand was not specified on the command, where *command_name* is the
command and *operand_name* is the omitted operand.

SLDS440E: TVISION The sensor specified on the STOP SENSOR command is not
started.
A stop Sensor command was issued for a Sensor that is not started.

SLDS441S: TVISION Unable to allocate the minimum size buffer queue.
Diagnostic data follow:
This error probably results from an installation limit on the size of data spaces. If not,
please contact Bristol Support and report the diagnostic data included in the message.

SLDS442W: TVISION The maximum number of allowed buffer queue blocks
specified on the MAXQBLKS startup parameter has been reached.
Events may be lost.
The Sensor required an additional queue block but the maximum allowed was already
allocated. When the current queue block is full and the maximum is still allocated,
events will be lost.

SLDS443I: TVISION Sensor quiescing: *n* events in buffer queue.
A stop Sensor command has been issued. The Sensor will wait until all events in the
buffer queue have been retrieved before completing the shutdown.

SLDS445I: TVISION Sensor quiesce completed.
All events in the buffer queue have been retrieved in response to a Sensor stop
command. The Sensor will complete the shutdown.

SLDS446E: TVISION The sensor manager component is not started.
This error may be caused by starting the Sensor driver manually. The Sensor driver is
automatically started by the Sensor manager and should not be started any other way. If
not the case, please contact Bristol Support.

SLDS448I: TVISION CICS CIC1 Sensor statistics:
Events in queue: *n*
Events collected: *n*
Events dispatched: *n*
Events_lost: *n*

Informational message issued at Sensor shutdown or in reponse to an INQUIRE
command.

SLDS449W TVISION Sensor events have been lost due to insufficient buffer
          queue storage.
          The Sensor manager issues this message the first time an event is lost.

SLDS451S: TVISION CICS *sysid* TransactionVision sensor stall condition
          detected. Diagnostic data follow:
          The Sensor manager is quiescing and detects that events are not being retrieved from
          the buffer queue by the Sensor driver, where *sysid* identifies the Sensor. The Sensor
          will exit quiesce mode and complete shutdown. Events in the buffer queue will be lost.
          Please contact Bristol Support and report the diagnostic data included in the message.

SLDS460I: TVISION Sensor exits manager started.
          Informational message issued in response to an SLDS transaction or PLTPI processing.

SLDS461I: TVISION Sensor exits manager waiting for sensor startup.
          The Sensor exits manager has been started but a corresponding Sensor manager has
          not.

SLDS462I: TVISION Sensor startup completed. Exits monitoring started.
          The Sensor exits manager has detected the startup of a corresponding Sensor manager.

SLDS463W: TVISION The sensor exits manager is already started. Request
          ignored.
          An SLDS transaction was invoked but the Sensor exits manager is already started.

SLDS464I: TVISION *exit_name* exit [enabled | disabled].
          The Sensor exits manager has enabled or disabled an exit, where *exit_name* is the name
          of the exit.

SLDS465I: TVISION Sensor exits manager terminated.
          An informational message issued when the Sensor exits manager completes shutdown
          as a result of an SLDP transaction, PLTSD processing, or the forced disablement of the
          program start exit.

SLDS466W: TVISION The sensor exits manager is not started. Request
          ignored.
          An SLDP or an SLDC transaction was invoked but the exits manager is not started.

SLDS467W: TVISION The request (*1st parm*) was not En(able) or Dis(able).
          An SLDC transaction was invoked but the first parameter was invalid.

SLDS468W: TVISION The request (*2nd parm*) was not: PS, KC, PC, IC, TD, TS,
          or FC.
          An SLDC transaction was invoked but the second parameter was invalid.

SLDS469I TVISION The program start exit (PS) cannot be disabled. The exit
         is stopped.
         An SLDC transaction specified disablement of the program start exit, which must
         remain enabled for the exits manager to run. The exit is stopped instead.

SLDS498S: TVISION [TransactionVision Manager | CICS *sysid* TransactionVision
          sensor] system error: message handler unavailable. Diagnostic
          data follow:
          Please contact Bristol Support and report the diagnostic data included in the message.

SLDS499S: TVISION [TransactionVision Manager | CICS *sysid* TransactionVision
          sensor] system error. Diagnostic data follow:
          Please contact Bristol Support and report the diagnostic data included in the message.

List of APIs Monitored

CICS Sensor

The CICS Sensor collects events for five groups of CICS APIs: file control, temporary storage, transient data, interval control, and program control. In addition, the CICS Sensor collects Task Start, Task End and Program Start events as well.

File Control APIs
APIs: WRITE, REWRITE, READ, STARTBR, RESETBR, READNEXT, READPREV, ENDBR, DELETE, UNLOCK

Temporary Storage APIs
WRITEQ TS, READQ TS, DELETEQ TS

Transient Data APIs
WRITEQ TD, READQ TD, DELETEQ TD

Interval Control APIs
DELAY, START, ATTACH, CANCEL

Program Control APIs
LINK, XCTL, RETURN

WebSphere MQ Sensor

The WebSphere MQ library, API exit and the z/OS WebSphere MQ Sensors collect events from the following MQ APIs.

- MQBACK
- MQBEGIN
- MQCLOSE

- MQCMIT

- MQCONN

- MQCONNX

- MQDISC

- MQGET

- MQINQ

- MQOPEN

- MQPUT

- MQPUT1

- MQSET

WBI (MQSI) Sensor

- MQSI2TRACE. This is an event generated by the TransactionVision trace and failure nodes for WBI.

WebSphere MQ-IMS Bridge Sensor

- MQIMS_BRIDGE_ENTRY

- MQIMS_BRIDGE_EXIT

These are events generated by the WbSphere MQ-IMS Bridge Sensor when a MQ message is received by the MQ-IMS bridge and when a reply is generated by the WebSphere MQ-IMS bridge.

Servlet Sensor

The following servlet classes are instrumented:

- javax.servlet.http.HttpServlet

- org.apache.jasper.runtime.HttpJspBase

- com.ibm.servlet.PageListServlet

The servlet Sensor tracks the following methods:

```
void doGet(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)
```

```
void doPost(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void doPut(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void service(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void doDelete(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void doHead(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void doOptions(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void doTrace(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void init()

void init(javax.servlet.ServletConfig)

void destroy()

void _jspService(javax.servlet.http.HttpServletRequest,
javax.servlet.http.HttpServletResponse)

void jspInit()

void jspDestroy()
```

Important! The TRACE and OPTIONS requests are different from the other requests because the servlet container has the following default implementation of these two requests:

- For TRACE requests, the servlet container returns the whole request string made by the client.

- For OPTIONS requests, the servlet container returns the list of HTTP requests handled by that particular servlet.

If you do NOT override the **doTrace** and **doOptions** methods, TransactionVision is unable to trace these requests, although the client will still get the response.

JMS Sensor

The JMS Sensor tracks the following methods:

```
Class:: TopicConnectionFactory
        createTopicConnection

Class:: QueueConnectionFactory
        createQueueConnection

Class:: Connection
        close
        start
        stop
        setExceptionListener

Class:: QueueConnection
        createQueueSession

Class:: TopicConnection
        createTopicSession

Class:: TopicSession
        createDurableSubscriber
        createPublisher
        createSubscriber
        unsubscribe
        commit
        recover
        rollback
        close

Class:: QueueSession
        createBrowser
        createReceiver
        createSender
        commit
        recover
        rollback
        createTemporaryQueue
        close
```

```
Class:: QueueSender
        send
        close

Class:: TopicPublisher
        publish
        close

Class:: QueueBrowser
        close
        getEnumeration

Class:: QueueEnumeration
        nextElement
        hasMoreElements

Class:: TopicSubscriber
        close
        receive
        setMessageListener

Class:: QueueReceiver
        close
        receive
        setMessageListener

Class:: Message
        acknowledge
```

EJB Sensor

The following EJB classes are instrumented:

- `javax.ejb.EntityBean`
- `javax.ejb.SessionBean`
- `javax.ejb.MessageDrivenBean`
- `javax.ejb.EnterpriseBean` (parent interface of all three above)

In addition to all business methods defined in public interfaces, the following methods will be instrumented:

- `ejbCreate()`
- `ejbPostCreate()`
- `ejbRemove()` (session/entity/message driven bean)

- `ejbLoad()` (entity bean)
- `ejbStore()` (entity bean)
- `onMessage()` (message driven bean)

Index

A

B

C

X