Getting Started Guide

Log Explorer®

Version 4.1 for Microsoft® SQL Server™ 2000 and 2005

LUMIGENT

Contents

Chapter 1
Log Explorer Introduction 1

Chapter 2
Installation 5

Chapter 3
Getting Started 17

About This Book

The *Getting Started Guide* provides conceptual information about the Lumigent® Log Explorer™ product (Log Explorer), as well as planning, installation, and usage information. This book defines terminology and various related concepts.

Intended Audience

This book provides information for individuals responsible for installing and using Log Explorer and understanding related Microsoft SQL Server concepts.

Other Information Available from Lumigent

Lumigent provides the following information resources:

Getting Started Guide

Provides conceptual information about Log Explorer, as well as planning, installation, and usage information.

Guided Tour

Provides a tour of Log Explorer's features, by guiding the user step-by-step in setting up a demonstration database and using Log Explorer against that database to solve database management, auditing, and recovery problems.

Help

Provides context-sensitive information and step-by-step guidance for common tasks, as well as definitions for each field on each window.

FAQ

Provides answers to frequently asked questions; available from the Lumigent website, www.lumigent.com

Conventions

This book uses consistent conventions to help you identify items throughout the documentation. The following table summarizes these conventions.

Convention	Use
Bold	• Window and menu items • Technical terms, when introduced
Italics	• Book and CD-ROM titles • Variable names and values • Emphasized words
`Fixed Font`	• File and folder names • Commands and code examples • Text you must type • Text (output) displayed in the command-line interface
Brackets, such as `[`*`value`*`]`	• Optional parameters of a command
Braces, such as `{`*`value`*`}`	• Required parameters of a command
Logical OR, such as *`value1`* `|` *`value2`*	• Exclusive parameters. Choose one parameter.

Contacting Lumigent

Lumigent Technologies, Inc., is dedicated to safeguarding the integrity and availability of corporate data. Please contact us with your questions and comments. We look forward to hearing from you.

To register your Log Explorer software, either register online at www.lumigent.com, or return the registration card enclosed in your product package. Benefits of registration include notification of product updates and upgrades.

For support around the world, please contact your local partner. If you cannot contact your partner, please contact our Technical Support team.

Telephone:	1 978 206-3677
Email (support):	support@lumigent.com
Email (sales):	sales@lumigent.com
Email (documentation):	support@lumigent.com
Email (product ideas):	support@lumigent.com
Web Site:	www.lumigent.com

Subscribers to the Log Explorer software maintenance and support plan receive product updates and unlimited priority technical support via phone or email for 12 months. This support covers a variety of issues, including installation and configuration, use of product features, and consultative assistance on using Log Explorer. For more information, contact your sales representative.

During a technical support interaction, you may be asked to provide the Product Version and File Version of your Log Explorer software. To access the Product Version and File Version, use the **Help > About** menu.

Chapter 1

Log Explorer Introduction

Managing Microsoft SQL Server can be a time-consuming task. Recovering data can be a difficult task that is critical to your business. The Lumigent Log Explorer product (Log Explorer) helps you address these problem areas.

What Is Log Explorer?

Log Explorer is the leading transaction analysis and data recovery solution for Microsoft SQL Server. By providing unprecedented access to the SQL Server transaction log, Log Explorer gives you the ability to understand and solve elusive database problems by:

- browsing the transaction log
- exporting data to create reports
- selectively recovering modified, deleted, dropped, or truncated data.

Application bugs and user errors can damage your data and your business. The ability to quickly find and recover from errors is critical to your business. Log Explorer enables online data repair and recovery for Microsoft SQL Server databases, avoiding costly and time-consuming offline techniques. In addition, a fine degree of granularity ensures that good transactions are not lost during recovery.

Log Explorer is tailored for the database administrator managing SQL Server in a production environment, offering full transaction log browsing, powerful data recovery, table salvage, and data export capabilities.

What Does Log Explorer Provide?

Log Explorer provides three key capabilities:

- Sophisticated *browsing* of transactional database activity
- Safe and flexible *selective data recovery*
- Filtered *log record data export*

Transaction Log Browsing

Use the powerful *browsing facility* to quickly understand application behavior:

- Browse the transaction log (online and backup) of your SQL Server database
- Filter the log information to provide customized views
- Select transactions based on time, activity types, tables, user IDs, SPIDs, and other criteria
- Examine the details of all database transactions and operations, including DDL commands and session login information
- View all data modifications for a specific table or a specific row
- View both user and system database activity
- Analyze load statistics based on the contents of the attached log
- Observe database transactions as they are added to the online log.

Selective Data Recovery

Use the flexible *data recovery facility* to easily recover from data loss or damage:

- Rollback a table to a specific point in time
- Rollback an individual row within a table to a specific point in time
- Undo a transaction, a sequence of transactions, or a single operation
- Reverse a series of deletions
- Redo a sequence of transactions
- Recover data from dropped or truncated tables.

The recovery facility uses a log-based rollback or redo approach. It does not use the traditional technique of restoring the database from a backup and rolling the transaction logs forward.

Recovery is *selective*. It repairs only those transactions and objects that you specify, avoiding the loss of good transactions and greatly reducing recovery time.

Finally, the recovery facility is *safe*. Log Explorer creates standard SQL scripts to undo bad transactions or recover lost data. You can review and verify the scripts before applying them to your live database.

Data Export

Use the flexible *data export facility* to easily export selected log records to the following targets:

- Files in XML, HTML, or character-separated-values format
- SQL Server tables

Use this exported data to generate a variety of reports using third-party or custom report generators.

Why Use Log Explorer?

Log Explorer transaction analysis and data recovery capabilities provide significant benefits to database administrators, system administrators, application developers, QA engineers, IT professionals, and corporate data auditors:

- Understand the internal behavior of your applications and SQL Server
- Significantly reduce recovery time
- Significantly reduce system downtime
- Recover lost or damaged data
- Repair selected objects without losing any unrelated transactions
- Repair data at various levels of granularity
- Quickly find application and user errors
- Verify application behavior
- Audit database changes
- Shorten the debugging cycle
- Analyze performance characteristics of your transactions
- Improve software quality.

Because Log Explorer uses the native transaction log already maintained by the database server, there is no runtime overhead. Unlike cumbersome trigger-based solutions that must be rebuilt whenever the application or database schema changes, Log Explorer operates with no impact on the server and no maintenance required.

How Does Log Explorer Work?

Log Explorer has a client component and a server component. The server component simply accesses the transaction logs and makes the log records available to the client component. The client component interprets filters, analyzes, and presents the log information to the user. In addition, the client component calculates SQL scripts for selective recovery and exports log data to files or SQL tables

Chapter 2

Installation

Log Explorer is licensed software. When you install Log Explorer, you will be asked to provide the product key that you received when you purchased the product. You will need that product key for upgrades, re-installation, and access to certain benefits offered to our customers.

Components and Requirements

Log Explorer consists of a Client component and a Server component. Using the Client component, you can attach to any SQL Server database on which the Server component is installed.

The setup program can install the Client component, the Server component, or both. It can also install the Server component on multiple machines simultaneously. The setup program is fully network-enabled; you can install either component from any machine on your network.

The Server component may reside on any system running SQL Server 2000 or 2005. The minimum machine configuration requirements for the Server component are the same as that for SQL Server. The Client component may co-reside with the Server component or may reside on a separate, networked machine running Windows XP, Windows 2000, or Windows Server 2003.

Before You Start

Before you start installing Log Explorer, you may find it convenient to gather the following information:

- The product key (see next section)
- The name of the server machine on which you want the Server component installed (if different from the installation system)
- The SQL Server user name and password for an account with administrative privileges on the server on which the Server component will be installed.

Licensing

The type of license that you have purchased for Log Explorer is reflected in the product key that you were provided along with your installation materials. You must enter this key in the appropriate field during installation of Log Explorer in order to enable it. Encoded in your product key is all the necessary information about your license.

Your license will be one of the following types:

- Single license – allows you to install and use Log Explorer with full functionality on one server only
- Multiple license – allows you to install and use Log Explorer with full functionality on multiple servers
- Subscription license – allows you to install and use Log Explorer with full functionality on multiple servers, but with a predefined expiration date (typically one year from purchase date). Beginning 60 days before the expiration date, the client will notify you of the impending expiration each time it is launched. On the expiration date, if you have not entered a new product key, the Log Explorer client will be disabled and will not attach to any database until you enter a new product key (using either the **Help|Enter Product Key** menu option or the Server Manager utility).

If you do not enter a product key during the initial installation of Log Explorer, you will automatically receive an evaluation license with a 30-day expiration period (thirty days from initial installation). This license allows you to evaluate Log Explorer with limited functionality; you may only attach to the Northwind, Pubs, and LumigentDemoDB databases. After the evaluation period has expired, you must purchase one of the license types described above in order to continue using Log Explorer.

Please note that this section does not constitute an actual license. Your license grant is displayed (in the Software License Agreement) when you install Log Explorer.

Installing Log Explorer on a Cluster

Log Explorer is not a cluster-aware application. Therefore, if you are running in a clustered environment (using Microsoft Cluster Service (MSCS) for Windows NT and/or Cluster Service for Windows 2000), you must install it on each active and passive instance of SQL Server. This applies only to the server component of Log Explorer; the client component may be installed anywhere.

Active nodes – Install Log Explorer on each active node using unique license keys issued at time of purchase.

Passive nodes – Install Log Explorer on each passive node using the active license keys provided above.

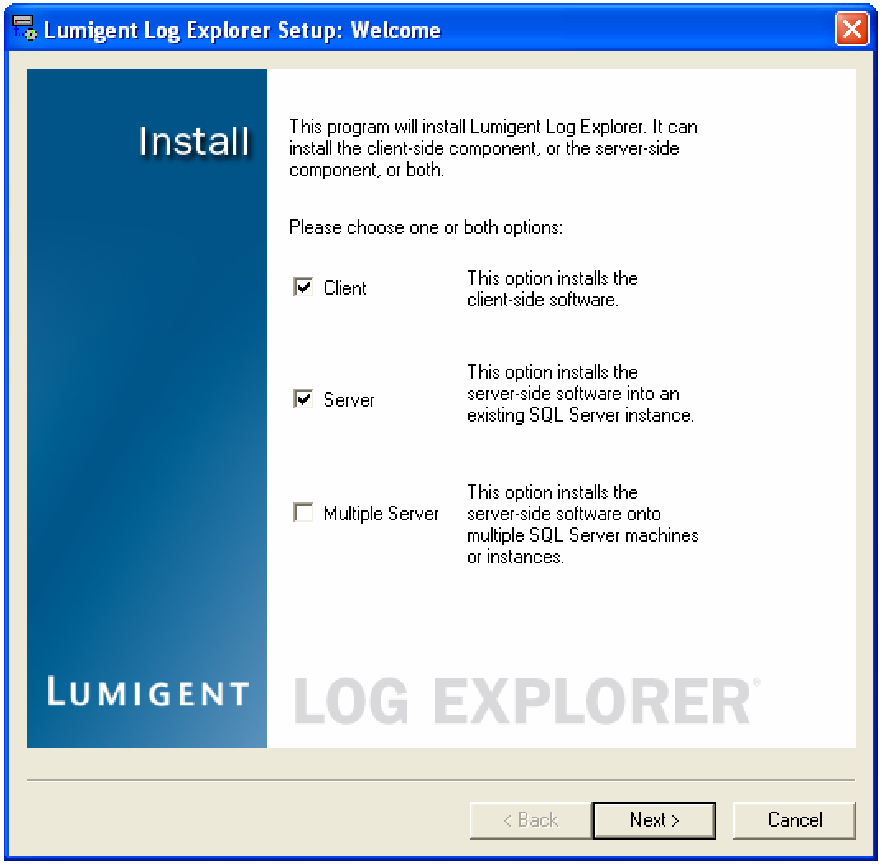
Note that the SQL Server service must be running on each node at installation time. If you need to install Log Explorer on a node with the SQL service stopped, please contact Lumigent customer support.

Client Installation

To install the Client component:

1. Client component selection step:

   Leave the **Client** check box checked in the **Install** pane. Click **Next**. (To install only the Client component without the Server component, unselect the **Server** and **Multiple Servers** check boxes.)



Lumigent Log Explorer Setup: Welcome

Install

This program will install Lumigent Log Explorer. It can install the client-side component, or the server-side component, or both.

Please choose one or both options:

☑ Client	This option installs the client-side software.
☑ Server	This option installs the server-side software into an existing SQL Server instance.
☐ Multiple Server	This option installs the server-side software onto multiple SQL Server machines or instances.

2. Product key entry step:

   Enter the product key you received with your licensed software, and click **Next**. To evaluate Log Explorer for 30 days (with certain restricted functionality), leave the product key field blank and click **Next**.

   Running Log Explorer in evaluation mode provides all the functionality of the full version, but access is limited to the `Northwind` or `pubs` sample databases (provided by Microsoft) or the `LumigentDemoDB` database (created by the Log Explorer Guided Tour).

   After 30 days, you will be notified that your evaluation period has expired, and you will no longer be able to attach to a log file. To discuss expanding your evaluation to include tests against your production or test databases, or to purchase a license to use Log Explorer, visit our Web site, or contact us at the email address or telephone number provided in this book.

3. License review step:

   Review the Log Explorer software license agreement and, if you agree to the terms, select the **I accept** check box and click **Next**. If you do not accept the terms, click **Cancel**, which will exit the installation process.

4. Choose Directory step:

   Identify the directory to use to install the Client component. The **Directory** field identifies a default directory on the computer on which the installer is running. To change the target directory, either click the **…** button for a directory navigation control or type directly in the **Directory** field.

5. Start Menu folder step:

   Choose a Start Menu folder on the installation computer where setup will install Log Explorer shortcut icons. Click the check box to have a shortcut to Log Explorer placed on the Windows desktop. Click **Next**.

6. (Optional) Server component installation step:

   If you are also installing the Server component, follow the instructions in the *Server Installation* section.

7. Installation completion step:

   Installation is now ready to proceed. Click **Next** to install the Client component, **Back** to return to a previous step, or **Cancel** to exit the installation process.

8. Review Release Notes. After installation is complete, you may want to read the Release Notes, which are available as an HTML document in the Log Explorer program group folder.

Single Server Installation

To install the Server component on a single server:

1. Server component selection step:

   Leave the **Server** check box checked in the Install pane and click **Next**.

2. Product key entry step:

   If you are installing the Server component without the Client, this leads to a product key entry step. Enter the product key you received with your licensed software, and click **Next**. To evaluate Log Explorer for 30 days (with certain restricted functionality), leave the product key field blank and click **Next**.

   Running Log Explorer in evaluation mode provides all the functionality of the full version, but access is limited to the `Northwind` or `pubs` sample databases (provided by Microsoft) or the `LumigentDemoDB` database (created by the Log Explorer Guided Tour).

After 30 days, you will be notified that your evaluation period has expired, and you will no longer be able to attach to a log file. To discuss expanding your evaluation to include tests against your production or test databases, or to purchase a license to use Log Explorer, visit our Web site, or contact us at the email address or telephone number provided in this book.

3. License review step:

   If you are installing the Server component without the Client component, this leads to a license review step. Review the Log Explorer software license agreement and, if you agree to the terms, select the **I accept** check box and click **Next**. If you do not accept the terms, click **Cancel**, which will exit the installation process.

4. Choose Server step:

   The **Server** field automatically contains the name of the computer on which the installation is proceeding. To install the Server component on a different computer, click the **…** button and select the target server, or simply type the name of the server into the text field.

5. SQL User Info step:

   Enter a user name and password for an account with administrative access rights to the server on which the Server component is to be installed. Select the check box if you want to use Windows authentication. Click **Next**.

6. Capture Login Information step:

   Check the **Capture login information** check box to tell Log Explorer to capture session login information, so that it can be correlated with activity in the log and displayed while you are browsing the log. Log Explorer stores the session information in a table called `leAuditCollectEventData`, in a database that you identify in this step. By default, this is the `lumigent` database, but you can change this by using the dropdown menu. Captured information is retained for 30 days by default. If you want the information to be retained for a different number of days, specify that number in the entry box.

7. Installation completion step:

   Installation is now ready to proceed. Click **Next** to install the selected components, **Back** to return to previous steps, or **Cancel** to exit.

8. Review Release Notes. After installation is complete, you may want to read the Release Notes, which are available as an HTML document in the Log Explorer program group folder.

Multiple Server Installation

To install the Server component on multiple servers at once:

1. Server component selection step:

   Check the **Multiple Server** checkbox in the Install pane and click **Next**.

2. Product key entry step:

   Enter the product key you received with your licensed software, and click **Next**. Note that for multiple-server installation, you must use a Site license key. If you use a Single license key, the installation will fail.

   To evaluate Log Explorer for 30 days (with certain restricted functionality), leave the product key field blank and click **Next**.

   Running Log Explorer in evaluation mode provides all the functionality of the full version, but access is limited to the `Northwind` or `pubs` sample databases (provided by Microsoft) or the `LumigentDemoDB` database (created by the Log Explorer Guided Tour).

   After 30 days, you will be notified that your evaluation period has expired, and you will no longer be able to attach to a log file. To discuss expanding your evaluation to include tests against your production or test databases, or to purchase a license to use Log Explorer, visit our Web site, or contact us at the email address or telephone number provided in this book.

3. License review step:

Review the Log Explorer software license agreement and, if you agree to the terms, select the **I accept** check box and click **Next**. If you do not accept the terms, click **Cancel**, which will exit the installation process.

4. If you are installing the Server component with the Client component, you will now see the Choose Directory and Start Menu Folder steps.

5. SQL User Info step:

Enter a user name and password for an account with administrative access rights to the servers on which the Server component is to be installed. Select the check box if you want to use Windows authentication. Note: The same login information must be valid for all servers on which you are installing. Click **Next**.

6. Choose Servers step:

There are three ways in which you can specify the servers on which to install.

- In the text box, type the name of a file containing a list of servers, or click the . . . button to browse to such a file. The file must be in text format, and each server name must be on a separate line.
- Click the "New" button and type the name of a server. Repeat as many times as needed.
- Click the **Find** button to bring up a dialog box displaying the names of all SQL servers found on the network. You can select multiple servers in this dialog by using shift-click or control-click.

7. Capture Login Information step:

Check the **Capture login information** check box to tell Log Explorer to capture session login information, so that it can be correlated with activity in the log and displayed while you are browsing the log. Log Explorer stores the session information in a table called `leAuditCollectEventData`, in a database that you identify in this step. By default, this is the `lumigent` database, but you can change this by using the dropdown menu. The same database will be used for each server on which you install Log Explorer. Captured information is retained for 30 days by default. If you want the information to be retained for a different number of days, specify that number in the entry box.

8. Installation completion step:

   Installation is now ready to proceed. Click **Next** to install the selected components, **Back** to return to previous steps, or **Cancel** to exit.

9. Review Release Notes. After installation is complete, you may want to read the Release Notes, which are available as an HTML document in the Log Explorer program group folder.

Enabling and Disabling the Capture of Session Login Information

Once Log Explorer is installed on a SQL Server, you can use the Server Manager tool to enable or disable the capturing of session login information on that server. The Server Manager tool is installed by the setup program in the same directory where you have installed Log Explorer. For more information on Server Manager, please see the tool's online Help.

You can also enable or disable session login information capture by using the `xp_logattach_SetCaptureState` extended stored procedure. The syntax of the command is as follows:

```
xp_logattach_SetCaptureState {1|0}, {retention},
   {'server-instance-name'}
```

`{1|0}`

Enables or disables the capturing of session login information on the server. `1` enables the data capture and `0` disables the data capture.

`retention`

Specifies the number of days that you want to keep the captured information in the `leAuditCollectEventData` table. When the information becomes older than the specified retention period, it is purged from the `leAuditCollectEventData` table.

`server-instance-name`

Specifies a named instance on which you want to have Log Explorer capture the session login information.

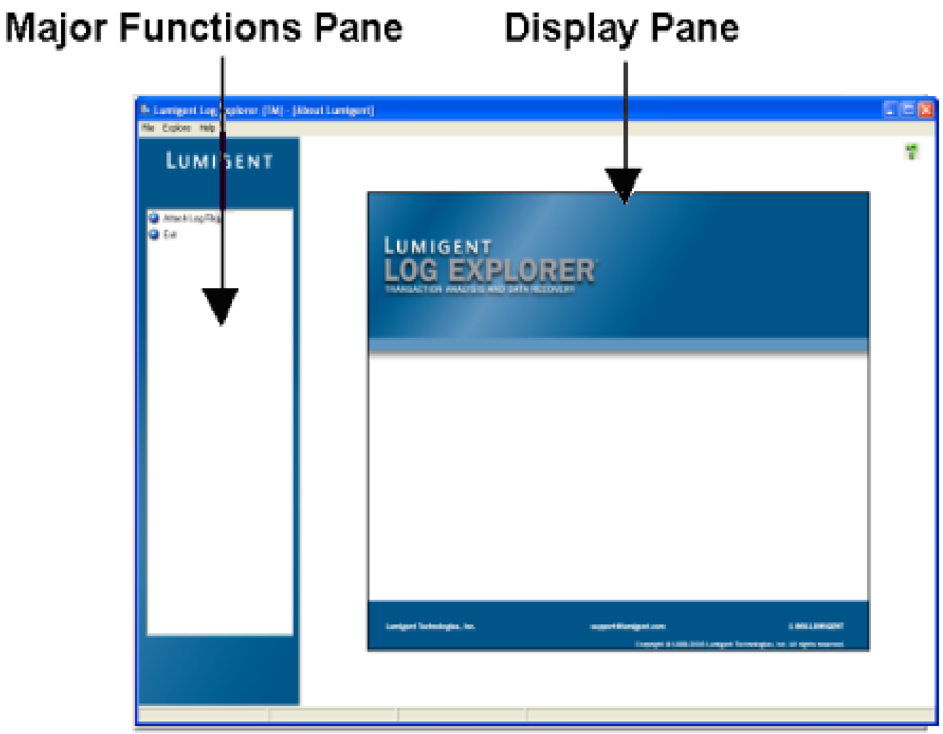
To capture login information, the SQL Server service account must be set up as a trusted login. This requirement is imposed by the SQL Server Profiler API.

Chapter 3

Getting Started

Start Log Explorer by double-clicking on the desktop icon or selecting Log Explorer from the Start Menu folder specified in the Client setup (by default, the Lumigent program group). You may also want to read the Release Notes, which are available as an HTML document in the Log Explorer program group folder.

The initial screen of Log Explorer shows the two panes that comprise the basic layout of the graphical user interface. You can resize the two panes by moving the splitter left or right.



For easy access to the online help, each Display pane shows the **?** button, which, when selected, displays the Log Explorer Help. You can also press your F1 function key to display help at any time.

Connecting to Your Database

The Major Functions pane initially offers two functions: **Attach Log File** and **Exit**.

When you click **Attach Log File**, you are prompted to enter (or select from a list) the name of the desired SQL Server instance, as well as login information to access it. (For SQL Server 2000, to access a non-default instance, identify both the server and the instance, such as `ServerName\InstanceName`.) You then click **Connect** and are prompted to type or select the name of the database whose logs you wish to view. At this point you can also use the checkboxes to select which log files Log Explorer should use; Log Explorer can use the online log and one or more backup log files. To identify specific backup files, click the **...** button for a file navigation control. Log Explorer will treat the set of selected files as a single virtual log.

Once you have finished specifying options, click **Attach**. The client will now attach to the target log files.

You may receive a warning that your log contains only recycled data (temporary data that SQL Server may overwrite as part of its normal operation). As the popup dialog notes, this indicates that one of the following conditions holds:

- Your SQL Server **Truncate Log On Checkpoint** option is set to true.
- Your database has never been backed up.
- Your log was truncated by `DUMP TRANSACTION WITH NO_LOG`.

For optimal use of Log Explorer, you should set the **Truncate Log On Checkpoint** option to `false`.

To change log files or databases at any time, use the **File > Attach Log File** menu entry. In general, you must re-attach Log Explorer to a database after making any change to its schema, such as re-creating a dropped table.

Attaching in a Cluster Environment

To attach to a SQL Server running on a Windows cluster, use the following format to specify the server name:

`\\Machine-Name\SQL-Server-Virtual-Name`

`Machine-Name`
Specifies the physical name of the Windows server that is hosting the active instance of your SQL Server.

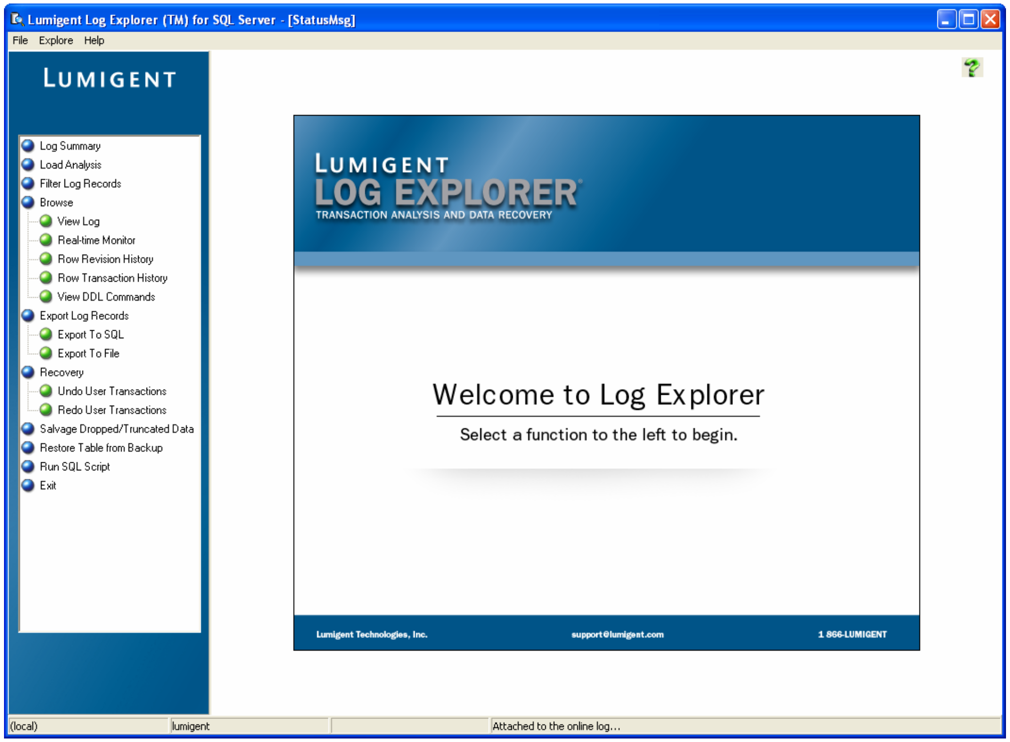
`SQL-Server-Virtual-Name`
Specifies the virtual SQL Server instance name.

Attaching through TCP

By default, the Log Explorer Client component connects to the Server component using named pipes. This is generally the best method if the client and the server reside on the same LAN in trusted Windows domains. If the client and server are separated by a WAN, or if they reside in completely separate Windows domains, it may be more effective to connect via TCP/IP. You can choose the desired protocol through the **File > Options** menu entry. Select the Protocols tab, select the appropriate radio button for the desired protocol, and click **Save**. The chosen protocol will be used next time you attach to a log file.

Log Explorer Functions

Once attached to a SQL Server log, Log Explorer offers the following major functions.



Welcome to Log Explorer

Select a function to the left to begin.

Attached to the online log…

	Description
Log Summary	Provides overview information about the attached log file.
Load Analysis	Computes measurements that describe the load against your SQL Server database.
Filter Log Records	Allows you to specify selection criteria that define a subset of log records, such as transactions, that are of interest to you. If you then invoke a browse, export, or recovery function, it will operate only on those transactions that are included in the selected subset.
Browse: View Log	Offers a rich set of functions for navigating a transaction log and viewing details of the transactions identified by the current filter.
Browse: Real-time Monitor	Allows you to view all database transactions as they are written to the log file. It is similar to a real-time transaction trace.
Browse: Row Revision History	Allows you to view the modification history for a specific table row, by reconstructing the values over time. It also allows you to revert the row to any of its previous revisions.
Browse: Row Transaction History	Allows you to list transactions and operations that affected a specific row in a specific table.
Browse: View DDL Commands	Allows you to view DDL commands that affect schema, SQL objects, and permissions.
Export Log Records: Export to SQL	Exports data from the log and stores it in a SQL Server table.
Export Log Records: Export to File	Exports data from the log and saves it as XML, HTML, or character separated values.

Recovery: Undo User Transactions	Allows you to reverse a set of selected transactions, for example, for recovering data that has been removed or modified incorrectly due to a user or application error.
Recovery: Redo User Transactions	Allows you to re-apply a set of selected transactions.
Salvage Dropped/Truncated Data	Allows you to salvage some or all data for a table that was dropped or truncated. Important! This feature does not support MS SQL 2005.
Restore Table from Backup	Important! This feature does not support MS SQL 2005.
Run SQL Script	Allows you to execute SQL commands stored in a script file.
Exit	Exits Log Explorer, maintaining your current settings (SQL Server, login name, database, etc.) as the initial values for your next session.

Log Explorer offers context-sensitive help; selecting the **?** button or F1 provides help about the currently active major function.

In the remainder of this section, we describe some common activities using Log Explorer and refer you to the online help for more information.

Exploring the Log

Log Explorer is designed to give you great flexibility in browsing the transaction log. A rich filtering capability allows you to identify precisely the information you want to see, and you can move easily among the browsing, filtering, export, and data recovery facilities.

Browsing

To browse the attached log, click **Browse** or **View Log** in the Major Functions pane. This will apply the current filter and present you with the Browse Log display pane. You can change the filter settings through the **Filter Log Records** major function. For more information, see the section on *Filtering*.

It is important to understand the nature of the log view that Log Explorer presents to the user. Log Explorer takes a *snapshot* of the log when attaching to the database, and it updates the snapshot upon each user refresh action (either an explicit Refresh or use of the real-time monitor function). The filter is applied to the snapshot, yielding the log view presented in the Browse Log display pane.

Understanding the Browse Log Pane

The main Browse Log pane shows the selected records in the designated time order (ascending chronological order by default). The records associated with each transaction id are identified with the same highlight color in the **TransId** column. The interleaved colors indicate interleaved operations of different transactions. The lower window displays the session login information and data values associated with the currently selected row (the row with the arrowhead cursor in the leftmost column of the record display). A splitter between the data value display and the session display permits you to control the amount of space devoted to each. You may move through the log records using mouse control of the scrollbar as well as familiar keyboard navigation means: Up arrow, Down arrow, Page Up, Page Down, Home, and End.

The Browse Log pane presents the following columns by default (in left-to-right order):

Column	Description
Cursor location	The current log record (identified by a right arrowhead).
Bookmark	Records that you have assigned bookmarks (indicated with an anchor image in this column) for easy navigation from any point in the log. To add a bookmark for a specific record, right click on that record and select **Add Bookmark**. To go to a bookmark, use the **Go To New Position** navigation icon or the **Browse > Go To** menu entry.
Time	The date and time that the associated transaction started. If the OpCode is `COMMIT_XACT` or `ABORT_XACT`, this shows the end date and time.
TransId	The transaction id of the record.
OpCode	A description of the operation captured by the record.
Table	The name of the table the operation acted on.
Index	The name of the affected index.
UID	The User ID that performed the operation.
SPID	The SPID of the process that performed the operation.
Desc	The user transaction name or an internal SQL Server description of the transaction.

You can change the displayed columns through the **File > Options** menu option. **Columns Available** indicates the columns not currently displayed that you may add—highlight a desired column name and click the right arrowhead icon. **Columns Selected** indicates the columns currently being displayed that you may remove—highlight a column name and click the left arrowhead icon. When finished, click **Save**. You may restore the default display settings by clicking the **Default** button. To control the left-to-right display order of the columns in the Browse Log view, first remove all of the columns from the **Columns Selected** area; then add columns from the **Columns Available** area in the desired order.

The toolbar available above the record display table in the Browse Log view offers a set of navigation control buttons; these provide a variety of view manipulation capabilities. Hover your mouse cursor over a button icon for a brief popup description of its function. From left to right, the navigation controls are:

Control	Description
Filter Log Records	Go to the Filter Log Records window.
Time Search	Position the log view at a specific time.
Go To New Position	Position the log view at a specific point in the log (using a bookmark, log sequence number, or relative movement).
Refresh View	Refresh the view of the transaction log since you last attached or refreshed. For example, update the snapshot to see any new transactions.
Previous Transaction	Find (for the currently selected log record) the nearest log record back in time that belongs to a different transaction.
Next Transaction	Find (for the currently selected log record) the nearest log record forward in time that belongs to a different transaction.
Top of Transaction	Find (for the currently selected log record) the first log record associated with the transaction to which the selected log record belongs.
Bottom of Transaction	Find (for the currently selected log record) the last log record associated with the transaction to which the selected log record belongs.
Previous Operation	Find (for the currently selected log record) the previous operation of the transaction to which the selected log record belongs.
Next Operation	Find (for the currently selected log record) the next operation of the transaction to which the selected log record belongs.

Control	**Description**
View Single Transaction	View (for the currently selected log record) all of the operations of the transaction to which the selected log record belongs, independent of any filter you may have set (operations belonging to other transactions will not be visible in Single Transaction mode but will again be visible according to the current filter when you toggle this button back into Multiple Transaction mode).
Reverse Time Sequence	Switch between viewing the log records in ascending or descending chronological order.
Enable/Disable Filter	Disable or enable filtering of log records.
Real-time monitor	View all database transactions as they occur in *real time*. This capability is similar to real-time transaction tracing. When placed in real-time mode, Log Explorer periodically checks for new transactions that have been appended to the on-line log file and displays them in this grid.
Switch to View1/View2	Switch between the two viewing contexts. You can have two independent viewing contexts (*View 1* and *View 2*) for browsing the log. For example, you can be at different positions within the two contexts. The current viewing context is indicated in the left-most status box on the bottom of the Browse Log pane.
Enlarge/Reduce Screen	Enlarge the amount of space devoted to log record display (the top descriptive graphic disappears) or to reduce the display space again.

In the Browse view, right-clicking on a row in the log record display presents a context-sensitive menu with the following entries:

Menu Entry	Description
Add Bookmark	Places a *bookmark* on the currently selected row. As you browse the log, you may want to return to a previously visited log record. A bookmark provides a convenient way to accomplish this task.
Clear Current Bookmark	Allows you to remove the bookmark associated with the currently selected log record.
Clear All Bookmarks	Allows you to remove all bookmarks.
Revision History	Shows you all revisions for the row that is affected by the selected log record. It is a shortcut to the **Row Revision History** major function.
View Current Row	Allows you to view the database row affected by the selected log record.
Undo Transaction	Allows you to undo all operations of this transaction. You can select a transaction by clicking on any of its operations.
Undo Operation	Allows you to undo a single operation. You can select an operation by clicking on the desired log record.

The File and Explore menus also offer access to Log Explorer major and minor functions.

Understanding Session Login Information

Directly underneath the main Browse View pane is the Session Login Information pane. When so enabled, Log Explorer captures session login information and keeps it in a SQL table called le`AuditCollectEventData`. This table resides in a database you select at install time.

To capture the login information, the server component of Log Explorer uses the standard SQL Server Profiler APIs. When a session connect event occurs, SQL Server passes the login information to the server component of Log Explorer. The server component of Log Explorer stores this information in the `leAuditCollectEventData` table.

The login information remains in the `leAuditCollectEventData` table for a period of time that you specified in the *`retention`* parameter at install time or at the time of invocation of the `xp_logattach_SetCaptureState` extended stored procedure. The server component of Log Explorer automatically deletes data from the `leAuditCollectEventData` table when that data becomes older than the specified retention period.

The Browse Log view displays the following session login information:

Entry	Description
NT User Name	The name of the Windows user that established the session associated with this record.
Login	The SQL Server login name that established the session.
NT Domain	The name of the Windows domain from which the session was established.
Application	The name of the application that established the session. This field is present only when the application specifically issues a SQL Server API that sets the name.
Client Host	The name of the Windows client from which the session was established.
Session Start	The time at which the session was started, in *`Year`*:*`Month`*:*`Day-Hour-Minute`*:*`Second`* format.

SQL Server may not always have all of the session information available. For example, in certain cases, SQL Server may not know the Windows user name or the domain name. The login information is available to SQL Server in the following cases: either the user established a connection using Windows authentication, or the client network protocol is set to Named Pipes. The login information is not available to SQL Server, due to SQL Server authentication, if the user established a connection using a non-trusted login and the client network protocol is set to TCP/IP.

Transaction Details

The bottom-most pane in the Browse Log window is the transaction details window. When a log record is highlighted in the main Browse View pane, this window displays the contents of the SQL table row affected by the transaction to which that log record belongs. If the transaction was a Modify, the transaction details window displays the column name and the Old and New values. Note that only the bits which have changed are displayed (e.g. if a string changed from “abc” to “abd” you will see “c” in the Old and “d” in the New column). To reconstruct the entire contents of the column both before and after the modify, click the **Reconstruct** button. You can also right-click on a row and select **Revision History** to see the full history of the row (see below under “Row Revision History”).

Filtering

The Filter Log Records function allows you to define the log records of interest. There are ten tabs in this pane:

Tab	Description
Time Range	Define the start point and end point. The start point may be the beginning of the log or a specific date and time in the log. Likewise, the end point may be the end of the log or a specific date and time in the log.
Recycled Data	Indicate whether you want to see recycled data within the log. Log records that have already been backed up may be reused at any time in the normal operation of SQL Server.
Activity	Indicate which log operation codes you are interested in. By default, the displayed log operation codes are INSERT_ROWS, DELETE_ROWS, MODIFY_ROW, MODIFY_COLUMNS, COMMIT_XACT, ABORT_XACT. You can also choose to view user data only or both user and system activity.
Table	Indicate the system and user tables of interest. Only log records for operations affecting the checked tables will be shown.
User Id	Indicate the User Ids of interest. Only log records for operations by the checked User Ids will be shown.

Tab	Description
SPID	Indicate the SPIDs of interest. Only log records for operations involving the given SPIDs will be shown. * indicates that all SPIDs will be matched.
Search Depth	Control how many records Log Explorer reads in matching your filter criteria. By default, Log Explorer will search all the records in the log. Reducing the search depth may result in shorter filtering times but may not match all the available records in the log.
Advanced	View transaction activity for a table that has been dropped. Log Explorer assumes that you have recreated the table with the original name and schema (the new table may still be empty).
Login Info	Indicate the Windows and/or SQL login IDs of interest. Only log records for operations by the checked usernames will be shown.
App Info	Indicate the hosts and/or applications of interest. Only log records for operations by the checked clients hosts/applications will be shown.

By default, the filter settings are not saved, so that when the user re-attaches, the filter will use the default settings. To save the current filter settings for use in attaching to the same database, check the **Save Filter Settings** option. If the next attachment is to the same database, the filter will use the saved settings (otherwise, the filter will use the default settings).

If desired, check the **Index Filtered Records** option to cause Log Explorer to create an index for the set of records described by the filter. On a large record set, this index may take some time to create, but once created it will make browsing the record set more efficient.

Real-Time Monitor

This mode allows you to view all database transactions as they occur in *real time*. This capability is similar to real-time transaction tracing. When placed in real-time mode, Log Explorer periodically checks for new transactions that have been appended to the on-line log file and displays them in the browse view. Log Explorer checks for new transactions periodically according to a refresh rate parameter, which is set to one second by default. Checking for new transactions involves reading the last data block of the on-line log file. This may create a minor performance overhead, depending on various factors, including the refresh rate.

To enter Real-time Monitor mode, click the **Real-time Monitor** navigation icon in the Browse Log view, or select **Real-time Monitor** from the Major Functions pane. You must be attached *only to the online log*. In Real-time Monitor mode, the Browse Log pane presents a toolbar through which you may perform the following functions:

Function	Description
Pause/Restart monitor	Causes Log Explorer to pause the real-time mode or to start checking again for new transactions.
Stop Real-time monitor	Causes Log Explorer to end the real-time mode and return to the normal browse mode.
Change Refresh Rate	Allows you to change the refresh period (in number of seconds) at which Log Explorer checks for the arrival of new transactions.
Enable/Disable Filter	Disables and re-enables the filter. If the filter is disabled, Log Explorer displays all new transactions that are written to the end of the online log. If the filter is enabled, Log Explorer displays only those new transactions that match all filter conditions that are currently in effect.
Enlarge/Reduce Screen	Enlarges the amount of space devoted to log record display (the top descriptive graphic disappears) or to reduces the display space again.

You may not scroll or view details of log records while in real-time monitor mode.

Row Revision History

The **Row Revision History** function allows you to view all modifications over time to a specific table row. This may be helpful for certain types of auditing or problem resolution that require an historical view of the data values.

The Row Revision History pane contains two tabs: Row Id and History. Use the Row Id tab to identify the table and row for which you want to see the revision history, by entering one or more column values. The History tab is the actual display area that shows the revision history for the selected row. The revision history is represented by showing the data row (by column values) as it existed at various points in time (prior to each modification). Row revisions are displayed in reverse chronological order, from the most recent to the oldest.

To create a row revision history:

1. Click on the **Row Revision History** function.
2. In the top portion of the Row Id tab, select the database table of interest from the **Table** pull-down list. Review the current time range; if you want to change the time range, click the **Time range** link or select the **Filter Log Records** major function.
3. In the bottom portion of the Row Id tab, select the column names of interest by clicking the **Include** checkboxes on the right; selected columns will be displayed in the result set. Enter column value selection criteria, by typing each desired column value into the corresponding row of the data entry grid. If this identifies more than one row, the **Seq** field can be used to indicate which of the matching rows should be selected.
4. Click **View** at the bottom of the screen to display the row revision history for the row that match the selection criteria.

You may also create a row revision history through the right-click menu of the Browse Log view.

Once you have created a row revision history, the History tab appears, allowing you to review it. Right-click on any revision to expose a menu for either rolling back the current row to the selected revision or browsing the transaction that created the selected revision. In addition, you may select an entry and click the **Browse** or **Rollback** buttons at the bottom of the pane.

Row Transaction History

The **Row Transaction History** function allows you to list transactions and operations that affected a specific row of a specific table. Unlike the **Row Revision History** function, this function does not reconstruct the entire row. It simply lists those transactions that affected the specified table row.

The Row Transaction History pane contains two tabs: Row Id and History. Use the Row Id tab to identify the table and row for which you want to list the transaction history, by entering one or more column values. The History tab is the actual display area that shows transactions affecting the selected row. Transactions are displayed in reverse chronological order, from the most recent to the oldest.

To create a row transaction history:

1. Click on the **Row Transaction History** function.
2. In the top portion of the Row Id tab, select the database table of interest from the **Table** pull-down list. Review the current time range; if you want to change the time range, click the time range link or select the **Filter Log Records** major function.
3. In the bottom portion of the Row Id tab, select the column names of interest by clicking the **Include** check boxes on the right; selected columns will be displayed in the result set. Enter column value selection criteria, by typing each desired column value into the corresponding row of the data entry grid. If this identifies more than one row, the **Row Seq** field can be used to indicate which of the matching rows should be selected.
4. Click **View** at the bottom of the screen to display the row revision history for the row that match the selection criteria.

In the History tab, review the row transaction history. Right-click on any transaction in the History view to go to the browse view.

View DDL Commands

The **View DDL Commands** function allows you to view DDL (Data Definition Language) commands. Unlike the DML (Data Manipulation Language) commands that alter user data (`INSERT`, `DELETE`, `UPDATE`), DDL commands create and destroy SQL Server objects (`CREATE TABLE`, `CREATE INDEX`, `DROP TABLE`, etc) and assign or revoke permissions (`GRANT`, `REVOKE`, etc). When you select this function from the Major Functions pane, Log Explorer reconstructs each DDL command from information in the transaction log within the currently active filter set. (For details on this process, see the Log Explorer online help.) This process can take a long time, so you may wish to narrow the field using a restrictive filter before you select this command.

The DDL Commands view is similar to the Browse Log view. The list of transactions is in the top pane, with details of the highlighted row in the lower pane. The toolbar above the transaction list contains three buttons:

Control	Description
Filter DDL Commands	Allows you to filter out DDL commands that are not of interest to you. This filter exists only for the **View DDL Commands** function and is not related to the filter found in the **Filter Log Records** function. Once you have created a filter with this command, you can toggle it on and off with the Enable/Disable Filter button (see below).
Refresh View	Causes Log Explorer to reattach to the log and re-display all DDL commands.
Enable/Disable Filter	Enables or disables the DDL filter.

At the bottom of the View DDL Commands window are three additional function buttons. Click **Redo** to generate a SQL script that contains commands to redo the highlighted DDL command. Click **Undo** to generate a SQL script that contains commands to undo the highlighted DDL command. Or, if the highlighted DDL command is a DROP TABLE command, click **Salvage** to re-create the dropped table and salvage the table's data. This function is also available from the **Salvage Dropped/Truncated Data** major function, described later in this book.

Right-clicking on a row in the View DDL Commands display presents a context-sensitive menu with the following entries:

Menu Entry	Description
Go to View Log	Takes you to the **Browse Log** major function and shows the rows associated with the selected DDL command.
Add to Dropped Table Filter	If the selected DDL command is a DROP TABLE command, this option will add the dropped table to a table filter. This will cause Log Explorer to include all transactions that occurred on the dropped table whenever Log Explorer applies a filter to select log records. For more information, see the Filtering section earlier in this chapter.
Copy	Copies the text of the DDL command to your clipboard.

Exporting Data

You can export data from the log in any of four formats: SQL Server table, XML, HTML, or character-separated values.

Export to SQL Server Table

To export the currently filtered log records to a SQL Server table:

1. Select **Export Log Records** or **Export To SQL** from the Major Functions pane.
2. Review the **Filter Summary** and **Filter Details** to verify the data to be exported. Click on an item in the Filter Summary window to see details of the selected item.
    a. If you want to change the filter settings, select the **Filter Details** title link or the **Filter Log Records** major function. After you have made your selections and clicked **Apply** (or clicked **Cancel**), you will be automatically returned to the Export major function. Click **Next** to continue.
    b. If you do not need to change the filter settings, simply click **Next**.

3. Select the destination SQL Server and click **Next**.

4. Identify the destination database: type the name of a database that contains the table in which you want to store extracted log records. The database must already exist on the selected destination SQL Server.

5. Identify the destination table: type the base name of a table in which you want to store extracted log records. If the table does not exist, Log Explorer will create it, along with four other tables called *`table-name`*`_details`, *`table-name`*`_keys`, *`table-name`*`_login`, and *`table-name`*`_ddl`, where *`table-name`* is the name you have selected.

6. Indicate through the corresponding check box whether to:

   Append this data to a pre-existing table

   Leaving this unchecked will erase any information previously stored in the destination table.

   Include detail data about each record

   Indicates whether to extract data contained in a log record and store them in the table called *`table-name`*`_details`, where *`table-name`* is the table name from the previous step.

   Reconstruct updates

   Leave this check box unselected to see delta changes exactly as they are recorded in the transaction log. Select this check box to see full update values. The reconstruction of full values involves extra disk IOs which may slightly degrade the performance of the system.

Show only committed transactions

Select this check box if you want to extract only committed transactions, omitting those transactions that have been aborted. Clear this check box if you want to extract committed as well as aborted transactions.

Stop On Error

Instructs Log Explorer on how to handle an error while exporting data (terminate immediately, or ignore and continue). The default is to stop on error.

7. Select **Finish**. This will result in the corresponding log records being written to the target database table.

Please note that the **Export to SQL** function cannot export log records that contain column data longer than 3000 bytes for any individual column.

Export to File

To export the currently filtered log records to a file:

1. Select **Export To File** from the Major Functions pane or use **File > Export Data**.
2. Review the **Filter Summary** and **Filter Details** to verify the data to be exported; click on an item in the Filter Summary window to see details. If you need to change the filter details, click the **Filter Details** title link or select the Filter Log Records pane. (When you click **Apply** or **Cancel** in the Filter Log Records pane, you will be automatically returned to the Export pane.)
3. Select the export format under **Export Option**. If you choose to export as character separated values, you may choose the delimiting character, or leave the default.
4. Select the target file name under **File Name**.
5. Select the maximum number of rows of data (log records) that you want to save. If you select **All Rows**, all records matching the current filter will be exported.
6. Select whether to create the target file in Unicode. The default is to create the target file using the ASCII character set.

7. Select a setting for **Stop On Error**, which instructs Log Explorer on how to handle an error while exporting data (terminate immediately, or ignore and continue). The default is to stop on error.
8. Select a setting for **View Export File**, which instructs Log Explorer to open the export file in the associated viewer application after creating it.
9. Click **Create**. This will result in the corresponding log records being written to the target file.

Recovering Data

One of the most exciting and powerful aspects of Log Explorer is its *selective data recovery* capability. You can recover from damage to your data while keeping your system online and without affecting unrelated data. Compare this to the traditional method of data recovery, in which one must take the system offline, install a backup of the database, and roll transaction logs forward to the point just before the data damage, losing the effects of any transactions after that point. The *backup and restore* technique may be fine for recovering from media failures, but it is not acceptable for dealing with application and user errors, that cause the vast majority of data damage.

Data recovery in Log Explorer comes in three main styles, each of which has its own major function:

- Undo User Transactions
- Redo User Transactions
- Salvage Dropped/ Truncated Data

Undo User Transactions

This function creates a script file containing standard SQL Insert, Delete, and Update commands that effectively reverse a sequence of transactions selected by the filter.

To undo user transactions:

1. Select **Undo User Transactions** or **Recovery** from the Major Functions pane.
2. Review the **Filter Summary** and **Filter Details** list boxes, which summarize the sequence of currently selected transactions; click on an item in the Filter Summary window to see details of the selected item. To change the filter, either select the **Filter Log Records** major function or click the **Change Filter** link. (When you click **Apply** or **Cancel** in the Filter Log Records pane, you will be automatically returned to the Undo pane.)

3. Identify the target file in which to store the SQL script for undoing the selected transactions. Type the filename directly into the entry field, or click the **…** button for a file navigation control.

4. Select a setting for **Stop On Error**, which instructs Log Explorer on how to handle an error while constructing the undo script. If the checkbox is selected, Log Explorer terminates the undo function as soon as it encounters an error. If the check box is cleared, Log Explorer ignores errors. By default, the check box is selected. You may want to clear the Stop On Error check box if you are recovering from a catastrophic failure and you would like Log Explorer to create a *best effort* undo script, in spite of any errors that may occur.

5. Select a setting for the **Include text, ntext, and image data** parameter. If this checkbox is selected, Log Explorer will include blob data in the Undo operation.

6. Select a setting for the **Retain Identity Values** parameter. When a column has the IDENTITY property, you may want to control whether the recovery script re-inserts the original identity values or whether it causes SQL Server to create new values. Select this check box if you want to retain the original identity values. This check box is selected by default. Clear this check box if you want the recovery script to generate new values for IDENTITY columns.

7. Select a setting for the **Do not restore column values that have been changed by subsequent modifications** parameter. If this box is checked, Log Explorer will skip columns whose values were changed after the user transaction(s) you are undoing.

8. Click **Create**.

This will cause an undo script to be written into the target file and displayed for your review. Log Explorer does *not* automatically execute the resulting SQL commands in the script. After reviewing the file, you can execute the script against the target database by using the Log Explorer **Run SQL Script** major function or the SQL Server Query Analyzer.

Redo User Transactions

This function creates a script file containing standard SQL Insert, Delete, and Update commands that effectively reapply a sequence of transactions selected by the filter. This can make it easy to reconstruct data even without a viable backup database and permits transactions to be copied and applied from one server to another.

To redo user transactions:

1. Select **Redo User Transactions** from the Major Functions pane. Review the **Filter Summary** and **Filter Details** list boxes that summarize the sequence of currently selected transactions; click on an item in the Filter Summary window to see details of the selected item. To change the filter, either select the **Filter Log Records** major function or click the **Change Filter** link. (When you click **Apply** or **Cancel** in the Filter Log Records pane, you will be automatically returned to the Export pane.)

2. Identify the target file in which to store the SQL script for redoing the selected transactions. Type the filename directly into the entry field, or click the **…** button for a file navigation control.

3. Select a setting for **Stop On Error**, which instructs Log Explorer on how to handle an error while constructing the redo script. If the check box is selected, Log Explorer terminates the undo function as soon as it encounters an error. If the check box is cleared, Log Explorer ignores errors. By default, the check box is selected. You may want to clear the Stop On Error check box if you are recovering from a catastrophic failure and you would like Log Explorer to create a *best effort* redo script, in spite of any errors that may occur.

4. Select a setting for the **Include text, ntext, and image data** parameter. If this checkbox is selected, Log Explorer will include blob data in the Undo operation.

5. Select a setting for the **Retain Identity Values** parameter. When a column has the IDENTITY property, you may want to control whether the recovery script re-inserts the original identity values or whether it causes SQL Server to create new values. Select this check box if you want to retain the original identity values. This check box is selected by default. Clear this check box if you want the recovery script to generate new values for IDENTITY columns.

6. Select a setting for the **Allow T-SQL Updates** parameter. Select this check box if you want Log Explorer to generate `UPDATE` commands in the redo script, when necessary. Clear this check box if you want Log Explorer to always generate an `INSERT` command instead of an `UPDATE` command. You may need to clear this checkbox if you are recovering data from a log file and the original database no longer exists. In this case, it would be incorrect to generate `UPDATE` commands, since you no longer have the original database rows to which the `UPDATE` command should be applied.
7. Select a setting for the **Include transactions for all tables that no longer exist in the database schema** parameter. Select this checkbox if you wish to replay transactions (INSERT, DELETE, UPDATE) for tables that no longer exist in the database. This option is useful when you are attempting to salvage as many transactions from the log as possible.
8. Click **Create**.

This will cause a redo script to be written into the target file and displayed for your review. Log Explorer does *not* automatically execute the resulting SQL commands in the script. After reviewing the file, you can execute the script against the target database by using the Log Explorer **Run SQL Script** major function or the SQL Server Query Analyzer.

Salvaging Data

Occasionally, you may lose all data in a SQL table due to a `DROP TABLE` or `TRUNCATE TABLE` command. Although SQL Server records `DROP` and `TRUNCATE` operations in the transaction log, it does not log the deleted data. Therefore, you cannot use Log Explorer's standard recovery functions to restore the deleted data from the log.

Log Explorer offers two mechanisms for salvaging dropped or truncated data: recover from free list, and restore from a backup file. Each mechanism is accessible from its own major-function window: **Salvage Dropped/Truncated Data** or **Restore Table from Backup**. (You can also access the Salvage function from the **View DDL Commands** window.) The Restore Table from Backup function can also be used to return a table to a particular point in time (i.e. to undo the effects of any INSERT or MODIFY statements as well as delete/truncate operations).

Salvage Dropped/Truncated Data

Important! This feature does not support MS SQL 2005.

If you do not have any backup files, then recovering from the SQL Server free list is the only option available to you for salvaging data. You can access this from the **Salvage Dropped/Truncated Data** major function.

When a table is dropped or truncated, SQL Server places all data pages that were holding the data on a free page list. If a page is not reused, it still holds the original table data. If you choose the free list recovery method, Log Explorer searches the free list for old data pages that have not been reused yet. It then creates a SQL script file that reconstructs the original data from these pages. Log Explorer is able to determine the number of original rows that have been dropped. On completion, it reports that number along with the number of rows that it actually recovered. This information indicates to the user whether dropped data was recovered fully or partially.

To salvage dropped or truncated data:

1. Select **Salvage Dropped/Truncated Data** from the Major Functions pane.
2. Select the name of the target table through the combo box, or type it if not listed.
3. Specify an approximate time at which the table was dropped or truncated. Log Explorer uses this time to start its search for a transaction that dropped the table. If you do not know this time, leave the default. The default time corresponds to the beginning of the log; this will result in a longer search time in most cases.
4. Select the appropriate radio button that distinguishes whether you want to recover data for a dropped table or a truncated table.
5. Identify the name of a file in which the Salvage Data function should store the resulting SQL script for restoring lost data.

6. Select a setting for **Stop On Error**, which instructs Log Explorer on how to handle an error while constructing the recovery script. If the check box is cleared, Log Explorer ignores errors. If the check box is selected, Log Explorer opens an edit window when it encounters an error, allowing you to proceed with the script or edit the offending line (see Run SQL Script later in this chapter). By default, the check box is selected. You may want to clear the **Stop On Error** check box if you are recovering from a catastrophic failure and you would like Log Explorer to create the *best effort* recovery script, despite any errors that may occur.
7. Select a setting for the **Include text, ntext, and image data** parameter. If this checkbox is selected, Log Explorer will include BLOB data in its operations.
8. Click **Create**.

This will cause a recovery script to be written into the target file and displayed for your review. Log Explorer does *not* automatically execute the resulting SQL commands in the script. After reviewing the file, you can execute the script against the target database by using the Log Explorer **Run SQL Script** major function or the SQL Server Query Analyzer.

Restore Table from Backup

Important! This feature does not support MS SQL 2005.

If you do have backups of your database, then you can restore the table directly from these backup files by using the **Restore Table from Backup** major function. You must have at least one full database backup; you can also use subsequent incremental database backups. Note that the backup(s) you use must have been taken before the DROP TABLE command, if any, because otherwise Log Explorer will report that the table no longer exists. (Similarly, if you want to restore the table to a specific point in time, make sure that you do not specify any backup files from after that point.)

Once you are ready to restore the table, simply enter the table name in the appropriate field in the **Restore Table from Backup** window and specify the full path to the backup file(s). Log Explorer automatically determines the chronology and the order in which to read the backup files.

To restore a table from backup:

1. Select **Restore Table from Backup** from the Major Functions pane.
2. Type the name of the target table in the appropriate box.
3. In the **Database Backup Files** box, use the **[…]** button to locate the relevant backup file(s).
4. Identify the name of a file in which Log Explorer should store the resulting SQL script for restoring lost data.
5. Select a setting for **Stop On Errors**, which instructs Log Explorer on how to handle an error while constructing the recovery script. If the check box is cleared, Log Explorer ignores errors. If the check box is selected, Log Explorer opens an edit window when it encounters an error, allowing you to proceed with the script or edit the offending line (see Run SQL Script later in this chapter). By default, the check box is selected. You may want to clear the **Stop On Errors** check box if you are recovering from a catastrophic failure and you would like Log Explorer to create the *best effort* recovery script, despite any errors that may occur.
6. Select a setting for the **Include text, ntext, and image data** parameter. If this checkbox is selected, Log Explorer will include BLOB data in the operations.
7. Click **Create**.

This will cause a recovery script to be written into the target file and displayed for your review. Log Explorer does *not* automatically execute the resulting SQL commands in the script. After reviewing the file, you can execute the script against the target database by using the Log Explorer **Run SQL Script** major function or the SQL Server Query Analyzer.

Recovery Tasks

The remainder of this section discusses how to perform some common recovery functions using the functions previously described:

- **Rollback** a table or a row to a specific point in time
- **Undelete** (restore) a set of deleted table rows
- **Undo** a specific operation or a specific transaction in Browse Log view.

Rollback a Table

Rolling a table back to a specific point in time is equivalent to undoing all user transactions on the given table from the target time to the present.

To rollback a table to a specific point in time:

1. Select the Filter Log Records major function.

2. On the Table tab, identify the table to be rolled back.

3. On the Time Range tab, set the **Start Point** to the target point in time to which the table should be recovered, and leave the **End Point** as the end of the log.

4. On the Activity tab, ensure all user activity codes are included. The user activity codes are `INSERT_ROWS`, `DELETE_ROWS`, `MODIFY_ROW`, `MODIFY_COLUMNS`, `COMMIT_XACT`, and `ABORT_XACT`.

5. Follow the instructions for *Undo User Transactions*.

Rollback a Row

Rolling back an individual row takes advantage of the **Row Revision History** function. This function permits you to select a specific row of interest from a particular table, by specifying column values that will identify that row. This is similar to doing a select on a table with a specific column value.

To rollback an individual row:

1. Click on the **Row Revision History** function.
2. In the top portion of the Row Id tab, select the database table of interest. Review the current time range. If you want to change the time range, click the **Time range** link or select the **Filter Log Records** major function.
3. In the bottom portion of the Row Id tab, select the column names of interest by clicking the boxes on the right. Checked boxes indicate columns that will be displayed in the result set. Enter column value selection criteria, by typing each desired column value into the corresponding row of the data entry grid. If this identifies more than one row, the **Seq** field can be used to indicate which of the matching rows should be selected.
4. Click **View** at the bottom of the screen to display the row revision history for the row that match the selection criteria.
5. In the History tab, review the row revision history and select a rollback target.
6. Click **Rollback** at the bottom of the screen, or right click on the target and select **Rollback Row**. In the same manner, you can also choose to jump to the Browse view with the pointer set to the selected transaction—either click the **Browse** button or select **Browse Transaction** from the right-click list.
7. Specify a target file name (or use the default) in the file chooser.
8. Click **Save** in the file chooser.

This will cause a rollback script to be written into the target file and displayed for your review. Log Explorer does *not* automatically execute the resulting SQL commands in the script. After reviewing the file, you can execute the script against the target database by using the Log Explorer **Run SQL Script** major function or the SQL Server Query Analyzer.

Undelete (Restore) Deleted Rows

Restoring deleted rows is accomplished by undoing the delete actions.

To restore deleted rows:

1. On the Activity tab, ensure only the **DELETE_ROWS** activity code is selected.
2. On the table tab, identify the tables on which the target deletions were performed.
3. On the Time Range tab, set the **Start Point** and **End Point** to identify the time range over which the deletions were performed. To undelete back to a specific point in time, set the **Start Point** to the target point in time to which the deleted rows should be restored and leave the **End Point** as the end of the log.
4. Follow the instructions for *Undo User Transactions*.

For even finer control over the undo operation, use additional filter settings. For example, to restore only the deletions performed by a certain SPID, set the SPID on the SPID tab.

Undo a Transaction in the Browse Log View

To undo a specific transaction, including all the operations of that transaction:

1. Use **Browse** to identify the desired transaction.
2. Right mouse click on the desired transaction.
3. Select **Undo Transaction** in the popup menu.
4. Specify a target file name (or use the default) in the file chooser.
5. Click **OK** in the file chooser.

This will cause an undo script to be written into the target file and displayed for your review. Log Explorer does *not* automatically execute the resulting SQL commands in the script. After reviewing the file, you can execute the script against the target database by using the Log Explorer **Run SQL Script** major function or the SQL Server Query Analyzer.

Undo an Operation in the Browse Log View

To undo a specific operation:

1. Use **Browse** to identify the desired operation.
2. Right mouse click on the desired operation.
3. Select **Undo Operation** in the popup menu.
4. Specify a target file name (or use the default) in the file chooser.
5. Click **OK** in the file chooser.

This will cause an undo script to be written into the target file and displayed for your review. Log Explorer does *not* automatically execute the resulting SQL commands in the script. After reviewing the file, execute the script against the target database by using the Log Explorer **Run SQL Script** major function or the SQL Server Query Analyzer.

Running a SQL Script

Each Log Explorer recovery function creates a script file that contains a sequence of standard SQL commands. You can execute this script by loading and running it from the standard SQL Server Query Analyzer window. If your script is relatively small, this approach may be sufficient. However, when the script contains a large number of commands, the standard Query Analyzer may not be appropriate. The main reason for this is that it is difficult to deal with individual command failures. First, if a command fails, it is not immediately obvious which command actually failed. Second, if you want to restart after a failure, you have to manually edit the script (removing all commands that executed successfully) and load it again.

The **Run SQL Script** function is designed to help you run a large script file. It offers flexibility and control that allows you to specify how you want to run the script and what you would like to do in the event of failures.

To run a SQL script:

1. Click on the **Run SQL Script** major function.
2. Identify the file containing the SQL commands to be executed. You may type a file path directly, or you may use the file explorer.
3. Indicate through the corresponding check box whether to:
    - Execute the sequence of commands as a **Single Transaction**: select the check box for a single transaction; leave the check box unselected for the default behavior, executing each command separately.
    - **Stop On Error**. Select this check box if you want to stop execution if an error occurs. This is the default setting. If an error does occur, Log Explorer opens up an edit window that displays the failed SQL command and gives you the following options:
        - **View/Modify Line** – View the line that caused the error. You can modify the line if desired, and then use the **Run Modified Line** button (see below).
        - **Ignore This Error and Continue** – Continue running the script from after the failed command
        - **Re-run Original Command and Continue** – Run the failed command again and then continue running the script
        - **Run Modified Line and Continue** – Run the script, but with a modified version of the failed line. Before you click this button, you must use the **View/Modify Line** button to make changes to the line.
        - **Cancel Run** – Cancel the script altogether and return to Log Explorer.
4. Click the **Run** button. The commands will be applied to the database to which you have attached.

The following command buttons are available:

Command	Description
View	This command opens the script file in Notepad.
Run	This command initiates execution of the script.
Cancel	This command terminates execution of the script.

As the script executes, Log Explorer automatically updates the fields **Succeeded**, **Failed**, and **Submitted**, which indicate how many commands from the script have succeeded, failed, and been submitted to this point. The fields **Insert**, **Update**, and **Delete** also indicate how many of each of those types of operations have been executed.