LOGTOOLS

USERS REFERENCE
for
LogTools Application

Written by Tony Ringsmuth

Version 5.0

Documentation Revision Date 09/07/2011

Table of Contents

TABLE OF CONTENTS **2**

PREFACE **3**

LOGTOOLS 4D VERSIONS COMPATIBILITY **4**

LOGTOOLS WILL NOT OPEN BOTH PRE-V11 LOGS AND V11+ LOGS AT THE SAME TIME. **6**

INSTALLATION AND SETUP **7**

LOG VIEWER **9**

TABLE VIEWER **11**

"COPY & EXPORT" DROP-DOWN MENU: 12
"SELECT" DROP-DOWN MENU 12

RECORD VIEW **13**

FILTERING **16**

FILE MENU **18**

OPEN LOG FOLDER 18
OPEN LOG FILE 18
CLOSE LOG FILE 18
REFRESH LOGS 18
MANAGE SCHEMAS 19
PREFERENCES 19
REGISTER LOG TOOLS 19
QUIT 19

VIEW MENU **20**

NEW LOG VIEW WINDOW 20
NEW TABLE VIEW WINDOW 20
USERS VIEW WINDOW 20
CONTEXT VIEW WINDOW 21

TOOLS MENU **22**

EXPORT SCHEMA 22
CLEAR FLUSH BUFFER ERROR 22
EXPORT LOGS 22
AUTOMATED LOG PROCESSING (NEW IN 3.0) 24
Log File State: *24*

WINDOWS MENU **27**

NEW FEATURES IN VERSION 3.0 ..... **28**

NEW FEATURES IN VERSION 4.0 ..... **28**

NEW FEATURES IN VERSION 4.1 ..... **28**

NEW FEATURES IN VERSION 5.0 ..... **28**

Preface

Since version 3.0 of 4th Dimension, with the inception of 4D Server, 4th Dimension has had the capacity to create backup and log files. While the Log file is potentially valuable in the case of a system crash and loss of data, 4D has provided very little capability to do anything else with the backup Log file. LogTools now fills this niche. Log tools gives you the ability to view operations from the 4D Log file.

LogTools 4D Versions Compatibility

LogTools is compatible with 4D Log file versions 3.5 to 2004, V11 and V12+. The actual structure of the 4D Log files varies greatly from pre-V11 Logs vs V11+ Logs. Therefore, LogTools shows the logs differently. Here are some of the more significant differences between them:

Pre-V11	V11+
Database Schema	
Database Schema is mainly stored in the header of the log file. LogTools can often know all table and field names directly from the log file.	No Database Schema in Log file. To see table & field names, the user must import a Schema to LogTools, or manually enter it by right-clicking on fields in LogTools.
Table Numbers & UUIDs	
Table information within each record is always references by 4D Table number	Table information within each record may have a mix of 4D Table numbers, and UUIDs (a 16 type universally unique ID). Create Record and Modify Record database operations are the only operations that have both a table number AND a UUID. Therefore, if a log file contains any Create Record or Modify Record operations, than LogTools learns and memorizes (in LogTools data file) the relationship between that UUID and that table number. All other database operations involving record only carry a UUID, not a table number.
Blobs, Pictures & Text Fields	
All portions of records are always stored as part of the record.	Portions of a record can be stored in separate database operations. Blob, Picture, and if specified in the database field properties, Text fields are stored separately from the parent record. These items will appear as a separate item in LogTools, because they really are a separate item in the data file. Note; If a parent record is updated, but it's blobs (or other large fields) are not updated, then, the associated sub-portions are NOT re-saved.
Subtables	
Subtable data is stored as part of the parent record, and will be displayed as part of the parent record.	Subtables are treated as a related table. When saving a parent record with subrecord, 4D opens an internal rollback transaction, saves the subrecord first, each

	as a separate operation, then saves the parent record, then validates the rollback-transaction. A special subtable key relates the subrecords to the parent record.
Process Numbers & Contexts	
Each database operation is saved with a Process number that reflects the 4D Task number (on the server) that performed the operation.	4D V11 introduces a new concept called a "Context". Beginning at data engine startup, the first database operation that alters records creates a new context. All operations that happen for that same user and process number remain in the same context for the lifespan of that process. At the beginning of a context, the 4D records into the Log file, the process number, user ID, machine name, and current machine owner. Each database operation within that context then records just the context ID. LogTools displays "Create Context" and a "Close Context" for these operations. A context ID is never repeated until the engine is restarted.
Database Operation IDs	
4D does not number each database operation. LogTools dynamically displays a "Log Sequence Number": (starting a 1 in each distinct log file) just for reference, but this is NOT in the actual log files.	4D record an "operation ID" with each database operation that is recorded in the log file. LogTools displays this number, as well as a dynamically generated "Log Sequence Number" just for reference.
Rollback Transactions	
4D only allows single level rollback-transactions. LogTools creates a "Transaction ID" for display, as a way to show and group all the items within a single rollback transaction together. When a record is created and saved within a rollback-transaction, it is given a temporary record number that begins at 18 million. If the transaction is validated, the record in the data file is assigned a permanent number, but the log file is never informed of that new number. Therefore, LogTools can not necessarily track the history of records created within rollback transactions.	4D allows multi-level rollback transactions. LogTools uses a different calculation to determine the resolution of each rollback transactions, and no longer uses the technique of a displayed "transaction ID". When a record is created and saved within a rollback-transaction, it is given a it's permanent record number. So, now we can finally relate this record to all other instances of the same record number.
Text Data	

Text data is stored is ASCII format.	Text data is stored in UTF-16 format.
LogTools 4.0 is still currently written in the 4D 2004 engine and does NOT natively deal with UTF-16 characters. When dealing with text data from V11, LogTools does a simplistic conversion: characters below 256 are converted directly to the corresponding ASCII character number. Characters above 255 are represented as "*<1234>*" where 1234 is any number above 255.	

LogTools will not open both pre-v11 logs and V11+ logs at the same time.

Installation and setup

LogTools is distributed as a simple zip file. Just un-zip the file.

On Macintosh:
Once un-zipped, you should have a single file: LogTools (application).

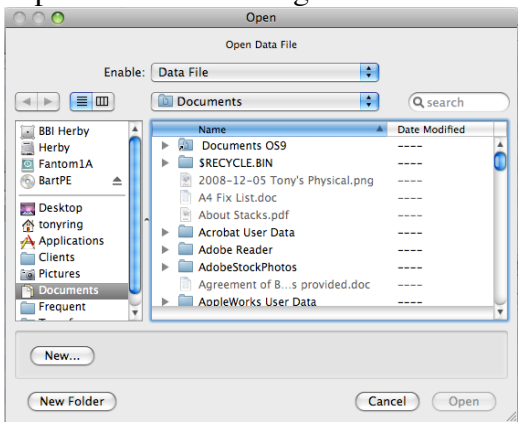
On MS Windows
Once un-zipped, you should have a "LogTools.app" folder. Within this folder, the LogTools.exe is your application.

The first time that you launch the application, you will be prompted to open or create a data file.

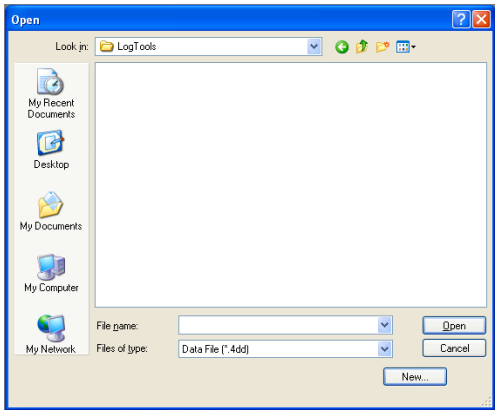
LogTools will use this data file to store
- your registration information
- imported schemas
- user preferences
- table UUID information

Click "New…" to create your LogTools data file.

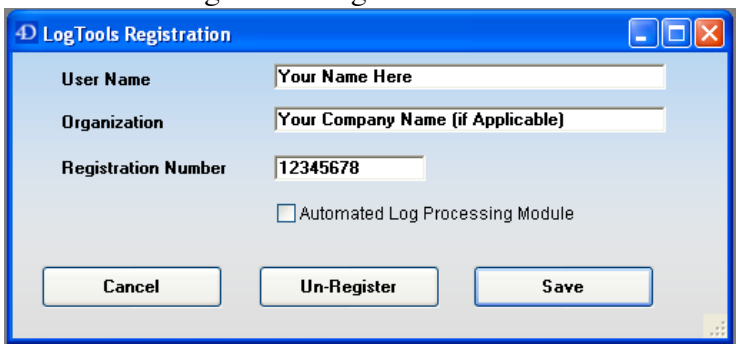
"open Data File" dialog on Macintosh



"open Data File" dialog on PC



At the first log-in to LogTools, you should enter your registration number (provided by BBI). If you have also purchased the Automated Log Processing Module, then check the "Automated Log Processing Module" check-box.



LogTools home window

Now that you're at the LogTools home screen, you can begin browsing Log Files immediately (File: Open Log File), or you can begin by importing a database schema (File: Manage Schemas)

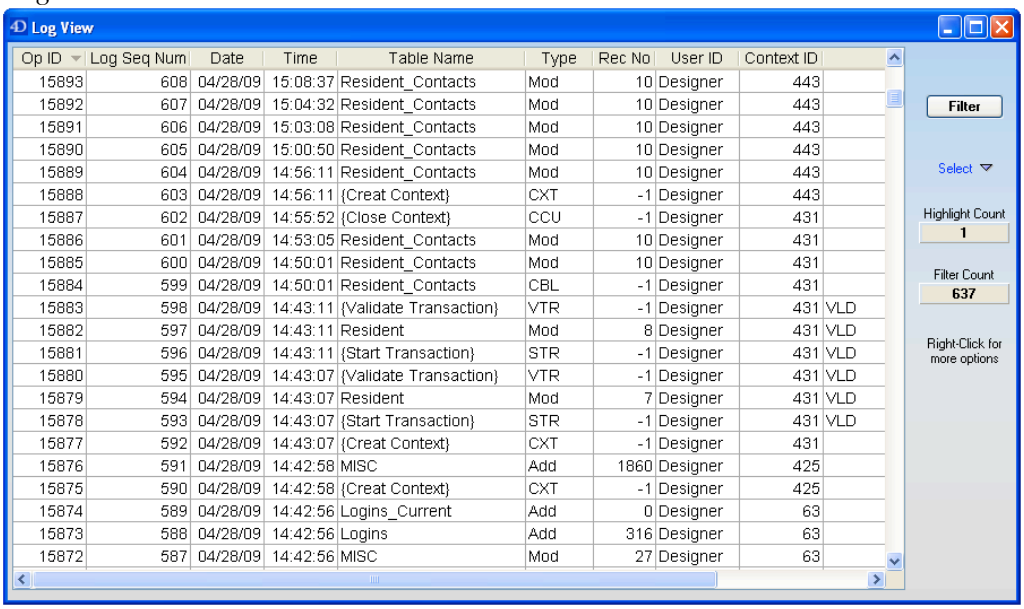
LogTools allows you to view 3 primary views of information from the Log file:

- Log Views
- Table Views
- Records

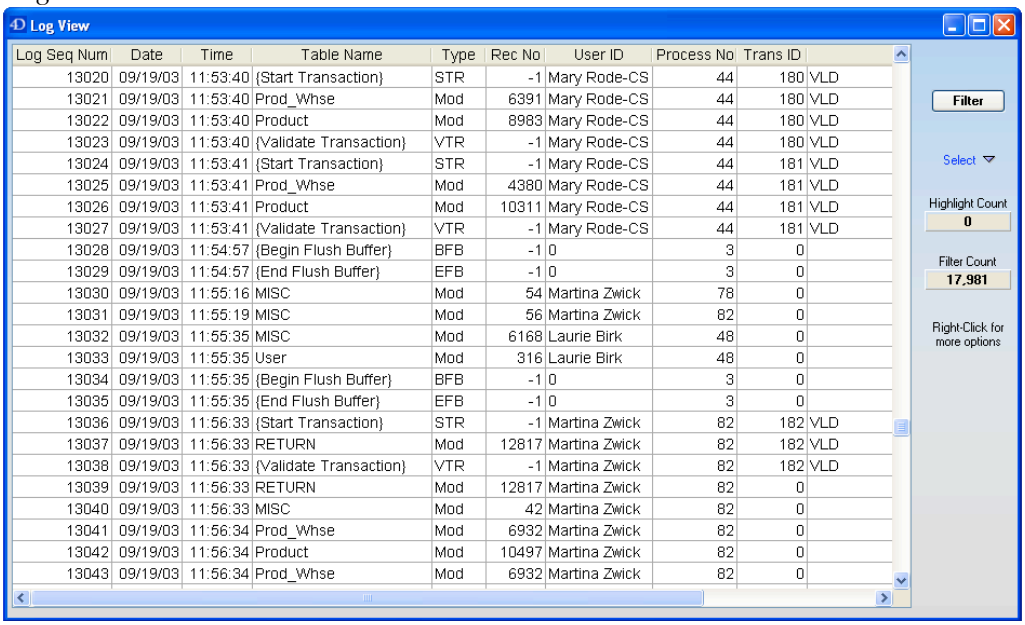
Log Viewer

{Menu} / View / New Log View Window

Log View Window: V11+



Log View Window: Pre-V11



A Log view is a view of a Log file. It allows you to see every transaction in the log file. You can sort or filter the view by any column. The log view allows you to see following columns:

- Operation ID (v11+): A unique ID assigned by 4D to each data operation that is entered into the Log file
- Log Sequence Number: A LogTools generated number: the sequence number of each log entry within it's log file
- {Activity} Date
- {Activity} Time
- Table Name (or number, or UUID, if the Table Name is not known)
- {Activity} Type (Add, Mod, Delete, etc)
- Record Number
- User Name or ID
- Context ID: (V11+) Identifies the context (derived from Process and User ID) for the operation.
- Process Number: The Process number on the database engine of the transaction.
- {Rollback} Transaction ID: (Pre V11) A sequential ID assigned by Log Tools to each rollback transaction. One rollback transaction can have one, none, or many activity items.
- {Rollback} Transaction resolution: How the rollback transaction was resolved: by Validate, Cancel, or incomplete.
-

Double click on any Add or Modify record to see a record view for the record.

The display rows are right-mouse clickable.

Filter Button: Click here to filter (search) the open log files on various criteria.

Select {drop-down list}:
The "Select" drop-down list is a context sensitive drop-down. Items include:

Open
Copy Columns
Copy Columns and Headers
Reduce Displayed List to Highlighted Items
Remove Highlighted Items from Displayed List
Highlight All In this Transaction
Highlight All In this Context (v11+)
Highlight All In this Process Number (Pre v11)
Highlight All for this Record
Highlight All for this User
Clear Highlighted Items from Log: Allows you to clear selected items out of the source log file: Caution when using this: this really does modify your log file, removing items from it. The purpose of this is that if you want to use 4D to integrate a log file, but you want to exclude some of the changes. Warning: you should only clear record modifications, not additions or deletions, otherwise a re-integration may become impossible. USE THIS ITEM AT YOUR OWN RISK.

Other Features:

You can sort the columns by clicking on the column headers.

You can have several open Log Views at a time. You open a new Log Viewer window by selecting the menu View: New Log View Window.

See Also: ***View: New Log View Window***

Table Viewer

{Menu} / View / New Log Table View Window

Table View Window, V11+

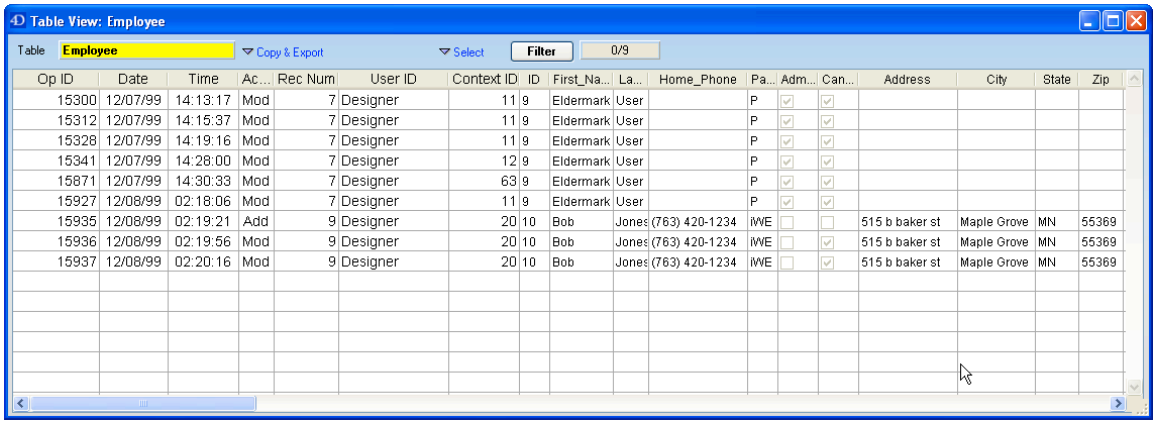
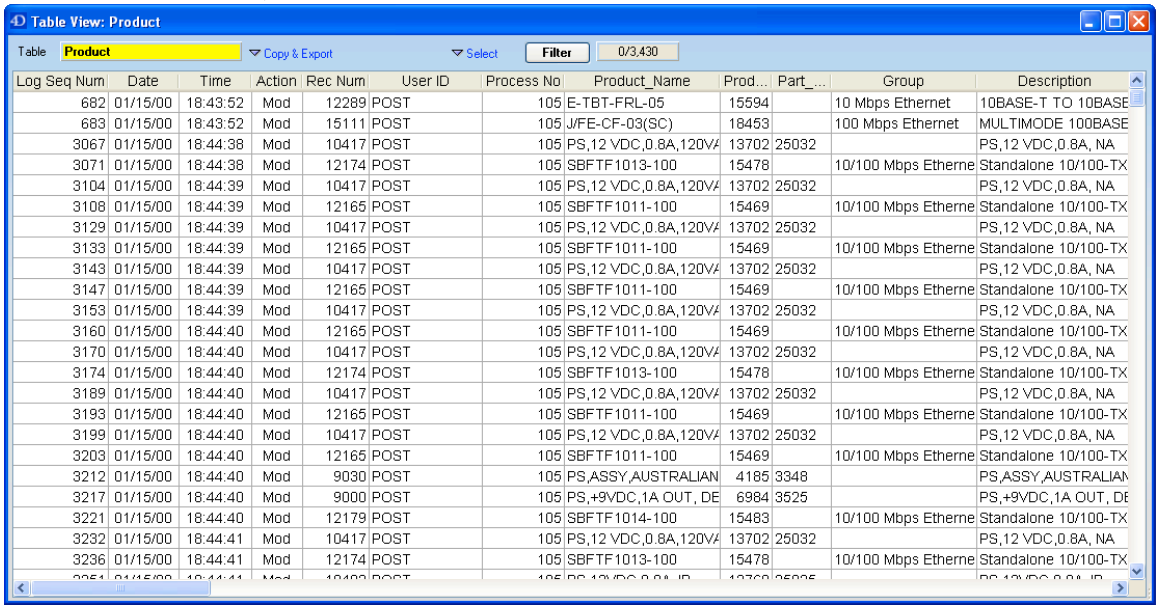


Table View Window, Pre-V11



A Table viewer is a view of records from a specific table from within the open logs. A Table View window displays the following columns:

- Operation ID (V11+)
- Log Sequence Number (Pre V11)
- {Activity} Date
- {Activity} Time
- Action (Add, Modify or Delete)
- Record Number
- User (Name or ID)
- Process Number (Pre V11) or Context ID (V11+)
- All the fields from the table

The power of the Table View window is that you can view all the fields in the table directly in the list view.

In the Table view, you can Filter on any of the fore mentioned columns and several others by clicking the Filter button. You can sort the columns by clicking on the column headers.

You can have several open Table Views at a time. You open a new Table View window by selecting the menu View: New Table View Window.

"Copy & Export" drop-down menu:

- ***Copy:*** Copies all highlighted row data, all columns
- ***Copy with headers:*** Copies all highlighted row data, all columns, with the column headers
- ***Save to Disk (CSV):*** Saves all highlighted row data for all columns, with headers to a CSV file that you create
- ***Save to Disk (Text):*** Saves all highlighted row data for all columns, with headers to a Text file that you create (New in version 3.0.3)
- ***Save to Disk (Hybrid CSV):*** Saves all highlighted row data for all columns, with headers to a hybrid CSV file (special characters get replaced)
- ***Copy Selective Columns…*** allows you to select the columns that you want to copy
- ***Copy Selective Columns & Headers:*** allows you to select columns that you want to copy, and includes header text in the copy

"Select" drop-down menu

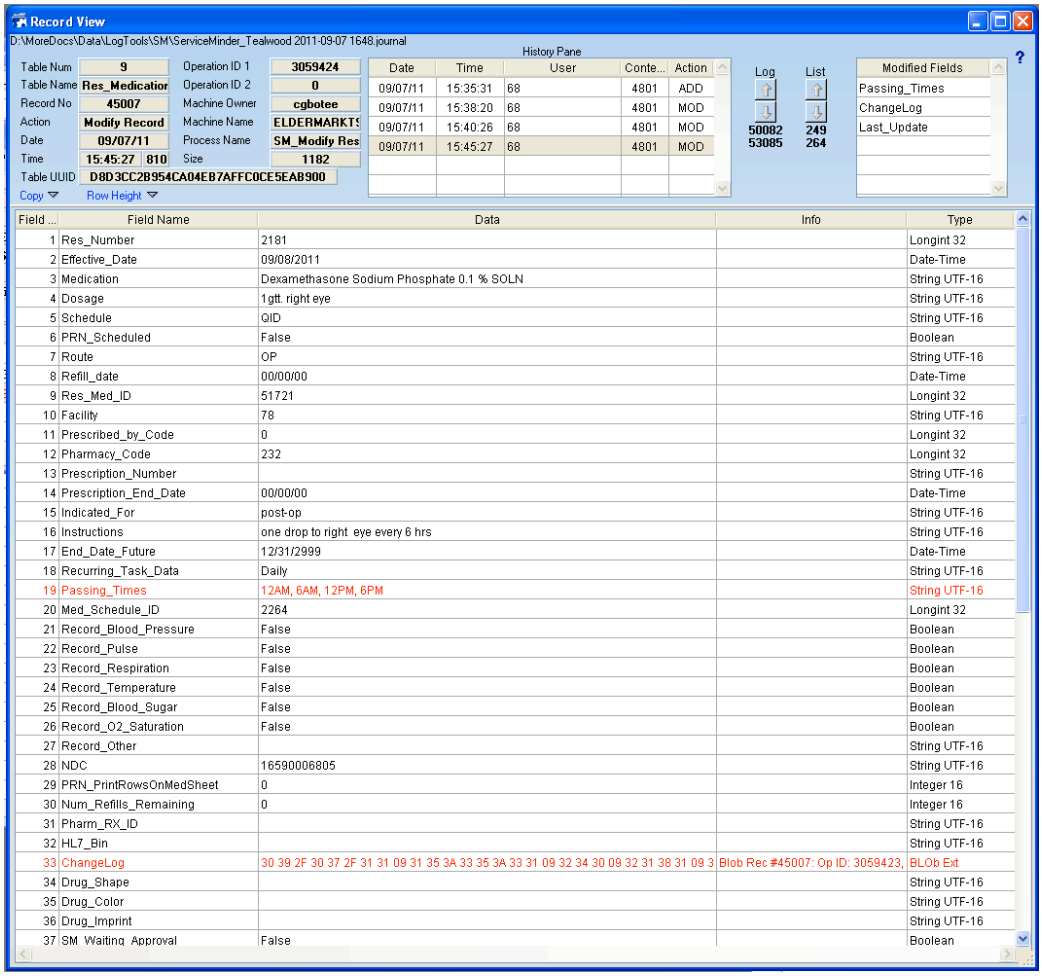
This drop-down menu allows you to keep or remove the currently highlighted rows from the viewed list of transactions.

See Also: ***View: New Table View Window***

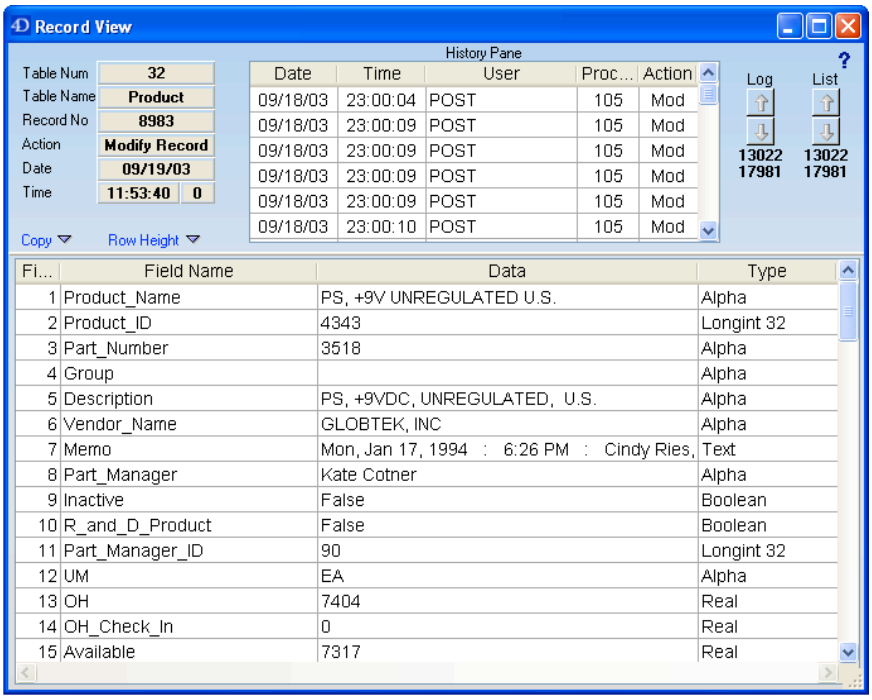
Record View

Double click on a row in the Log View or the Table View window to open a Record View.

Record View Window: V11+



Record View Window Pre-V11



- A Record view is a view of one specific record. You get to a Record Viewer window by double clicking on a Add or Modify record activity item in a Log View window or a Table View window or in the history pane of a Record View.

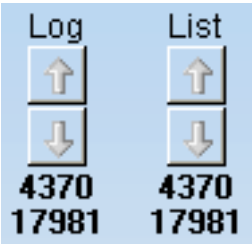
The history pane shows the modification history for this record within all open logs. Click on any item in the history pane to load that instance of the record into the Record View. Double click on an item in the history pane to open the item in a new window.

In the data columns of this window, fields that have been modified since the prior version of this record are displayed in red.

You can double click on fields of type Text, Blob or Picture or Subfield to see the full content of the field.

Right-mouse click on any row for more options, including entering/updating field names.

Next/Previous buttons:



The Next/Previous **Log** button loads the next or previous record from within the open log files. The numbers beneath the button show

- your current browse position within all operations

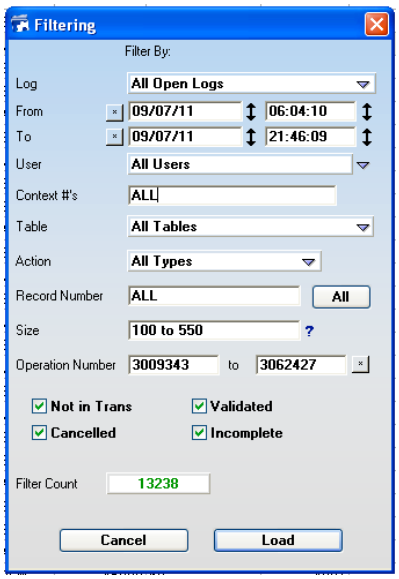
- the total number of operations open

The Next/Previous **List** button loads the next or previous record from the list of items from which you opened this Record View window. The numbers beneath the button show

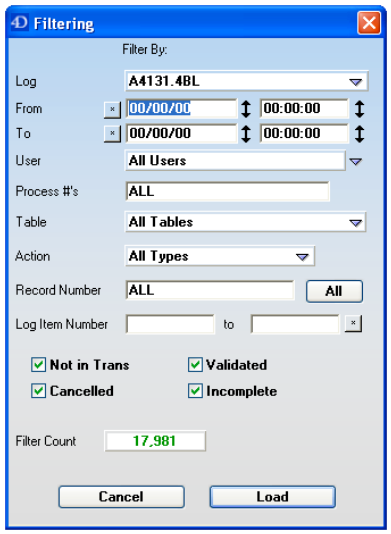
- your current browse position within the list that spawned this record view window
- the total number of operations in the list you came from

Filtering

V11+



Pre-V11



Filtering is similar to a querying: it allows you to limit your log view or table view to a selected criteria. You can Filter from either a Log View or a Table View.
You can filter on the following items:

- Open Log Files
- Date and Time Range
- User(s)
- Process Numbers or Context IDs
- Tables
- Action Type (Add, Mod, Delete, etc)
- Record Number{s}
- Log Item Numbers (Pre V11), or Operation IDs (V11+)

- Items based on their rollback transaction resolution

As you enter the criteria for your filter, the Filter Count at the bottom of the window displays how many log entries match the criteria.

FILE Menu

Open Log Folder

Open Log Folder allows you to select a folder from which Log Tools will open all enclosed log files (only at the root level of the folder).

Open Log File

Select *Open Log File* to open either a 4D Log file or a 4D Log Backup file. You can have multiple open log files at any given time. Generally, the intention in having multiple open logs at once is to open multiple, sequential log files from the same database to view activity across the multiple logs.

While you can open multiple log files from different databases at once, LogTools only manages one common set of table and field names for all open log files. It is intended that you open log files from the same database.

LogTools opens the Log file in read only mode. You can open the log file of a database that is currently running and recording transactions. Log tools does NOT currently update the log file with new transactions as the log file grows: you need to close and re-open the log file.

After you open a Log File, Log Tools will open a Log View window and display the newly opened log, if the log has less than 32000 transactions.

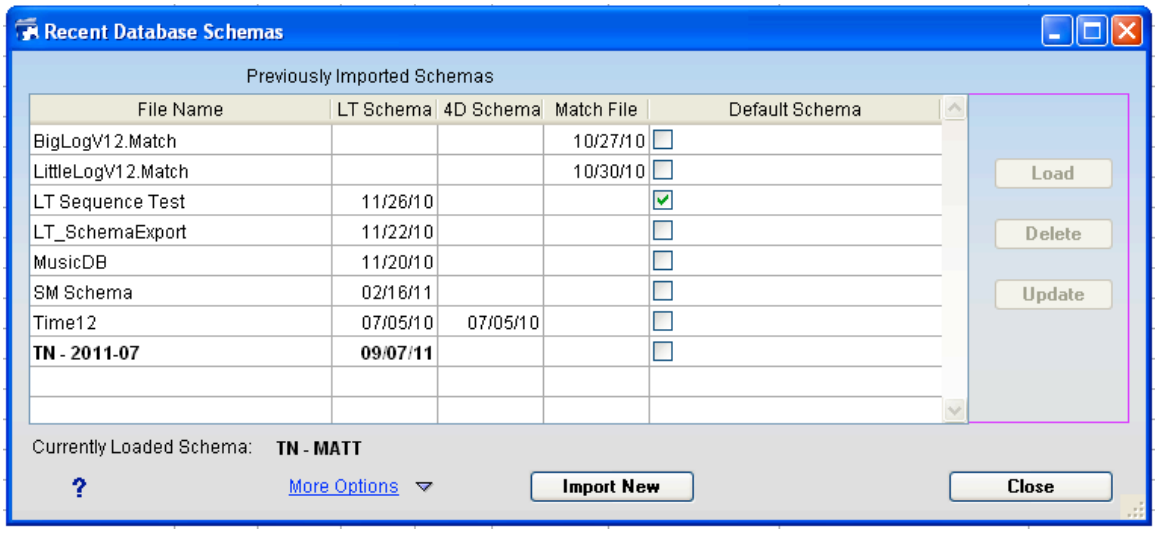
Close Log File

Select *Close Log File* to close any currently opened log file.

Refresh Logs

Refresh Logs scans all open log files for new transactions added to the end of each log file. This item is useful only in log files that are currently being written to. When you refresh the log, if any new transactions are found, they are added to the log data in memory, but are NOT immediately incorporated into any Log Views or Table View windows. To view the new transactions in any open Log View or Table View window, you must re-filter within the window.

Manage Schemas



This window allows you to import and manage loaded Schemas for the various database log files that you may browse.

If you have access to your 4D design environment, then you can create a method to export a LogTools compatible schema document to import here. (click the More Options drop-down).

You can also set any of the saved schemas to be a "On Startup Default Schema" which will automatically load every time that you start LogTools.

Preferences

Preferences allows you to set the Font and Font Size of the display areas.

Register Log Tools

Register Log Tools allows you to enter the registration number for your copy of LogTools. LogTools can run in demo mode without a registration number and will limit the number of log entries that can be viewed.

Quit

Select *Quit* to quit from LogTools. It is safe to Quit without closing any open log files.

VIEW Menu

New Log View Window

Select New Log View Window to open a new Log View Window.

The Log View Window shows a view the data operations of log file

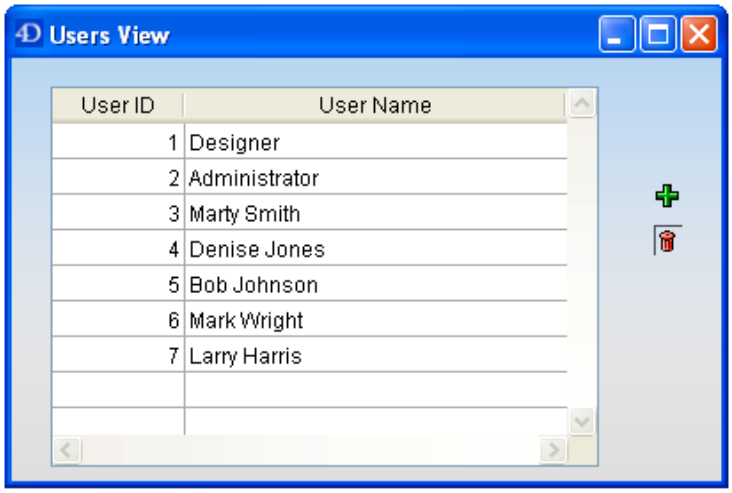
New Table View Window

Select New Table View Window to open a new Log View Window.

The Table View Window data operations for one specific database table, with columns for every field in the table.

Users View Window

Picture of the Users View window

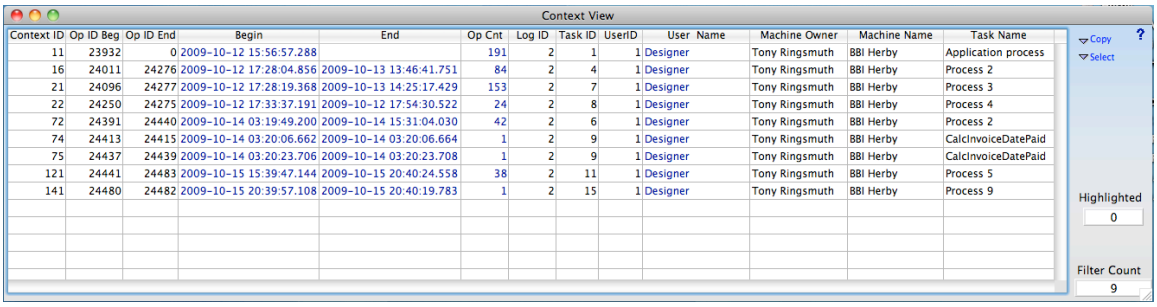


The purpose of this window is to
- Allow you to see the defined list of users that LogTools is using
- Allow you to edit the list so that in the record viewing windows, you will have correct user name displays.

Double click to edit the rows here.

Context View Window

(new in LogTools v4.1)
{Menu} / View / New Context View Window
(for V11 Log files only)



4D V11 introduces a new concept called a “Context”. Beginning at data engine startup, the first database operation that alters records creates a new context. All operations that happen for that same user and process number remain in the same context for the lifespan of that process. At the beginning of a context, the 4D records into the Log file, the process number, user ID, machine name, and current machine owner. Each database operation within that context then records just the context ID. LogTools displays “Create Context” and a “Close Context” for these operations. A context ID is never repeated until the engine is restarted.

The Context View window displays a listing of all Contexts that are found in currently opened logs.

TOOLS Menu

Export Schema

Export Schema exports a text format of the database schema for the current open log file. This export can be re-imported into LogTools.

Clear Flush Buffer Error

This is a Pre-Version 11 utility. This utility allows you to bandage a data file that has a "Interruption while flushing data to disk" error.

How to use:

- Select the menu item: Tools: Clear Flush Buffer Error
- Log Tools will prompt you to open a 4D Data file: Locate and open your damaged 4D data file.
- Finished.

Caution: While 4D will let you use the data file again after this "Band-Aid" utility has been run, your data file may have serious damage. It is a very good idea to repair the file with 4D tools, or check the data file with Datacheck.

Export Logs

Export Logs exports all currently open log files into a CSV or Text file that you name.

The columns of the export file are as follows:
Operation ID (V11+)
Log Sequence Number: the sequence number within the logs
Date: The activity date for this row
Time: The activity time for this row
Table Number: for activity items that deal with a record, this is the 4D table number of the record.
Table Name: for activity items that deal with a record, this is the 4D table name of the record.
Action Type: Every 4D log event has an activity type.

ADD	Add Record
DEL	Delete Record
MOD	Modify Record
STR	Start Transaction
VTR	Validate Transaction
CTR	Cancel Transaction
OPN	Open the Database
QIT	Quit Application

BFB	Begin Flush Buffer
EFB	End Flush Buffer
DED	Dead Space
CTB	Create Context
CTE	Close Context
BLC	Create Blob
BLM	Modify Blob
BLD	Delete Blob
BKS	Start Backup
TRT	Truncate Table

Record Number: the 4D record number involved for this action. For activity items that do not involve a record number such as Start Transaction.
User ID
User Name
Context ID (V11+)
Process Number
Rollback Transaction ID (Pre-V11)
Rollback Transaction Resolution
Field 1
{Field 2-N}

Subtables can be exported either as a column within the parent file, or as a separate file. When exporting as part of the parent file, the format is approximately this (a loose xml format):
<SubRecs count={n}>
<SubRec number={x}>Field1 *tab* Field2 *tab* Field3 <\SubRec>
</SubRecs>

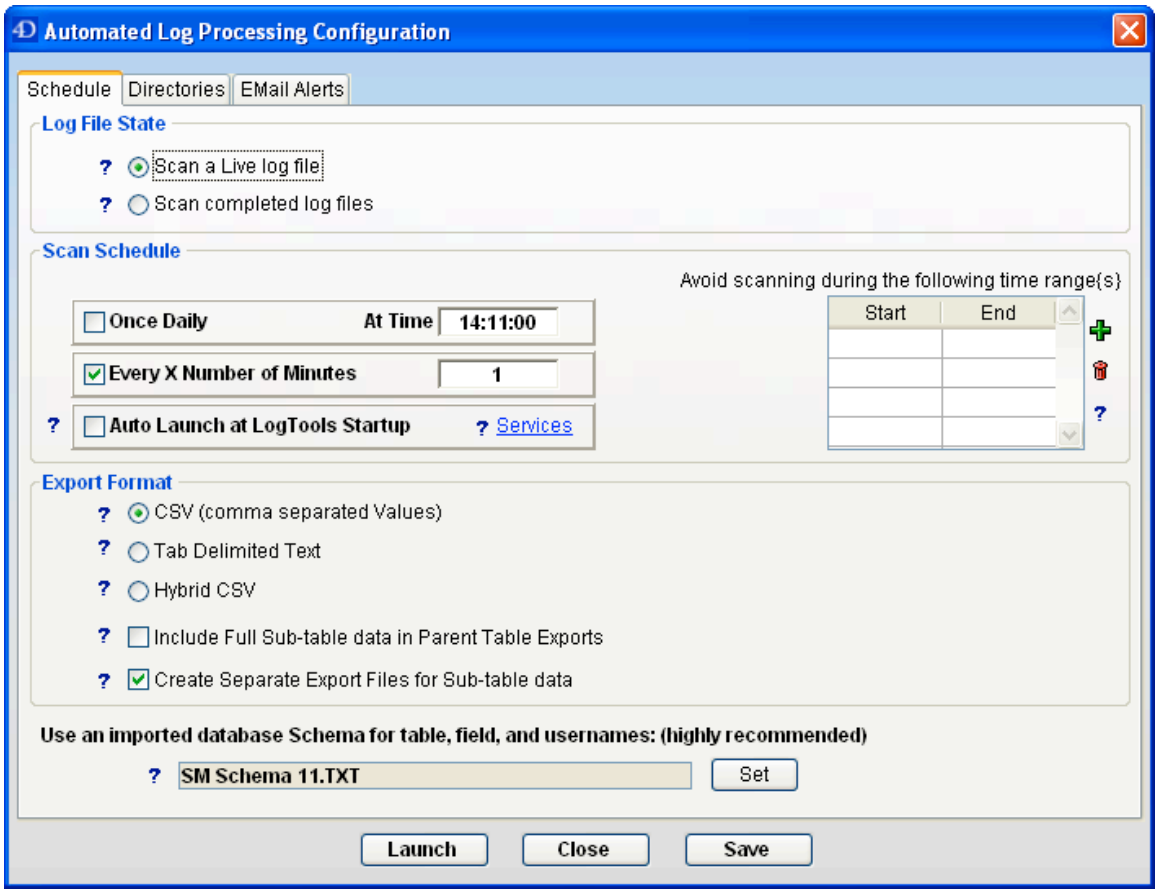
Within this sub record format, any embedded tabs are converted to “<TAB>” and carriage returns to “<CR>”.

Automated Log Processing (new in 3.0)

Automated Log Processing is a feature that allows LogTools to run continually, to:

- Scan for new Log Backup Files
- Process and Export them
- Move the Log Backup File to an alternate directory
- Send Email Notifications

As part of this mechanism, LogTools can register it's self (on MS Windows) to run as a service.



Log File State:

Scan a Live Log File:

The 'Scan a Live log file' option is for scanning the live log file of a running database.

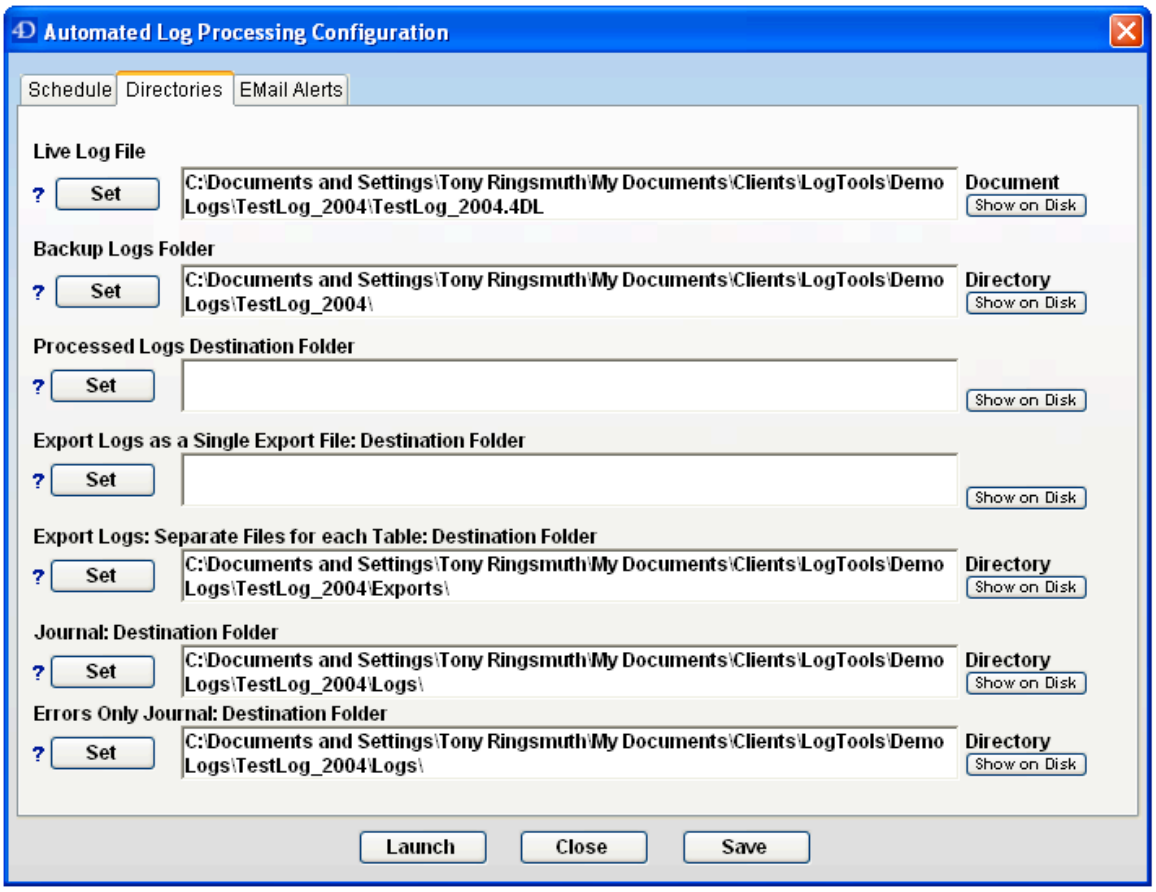
LogTools will continually scan you live log file once every X number of minutes, and append data to the export files.
When 4D starts a new log file, LogTools will start a new batch of export files. The export files will be named with a date/time stamp corresponding to the start time of the log file.

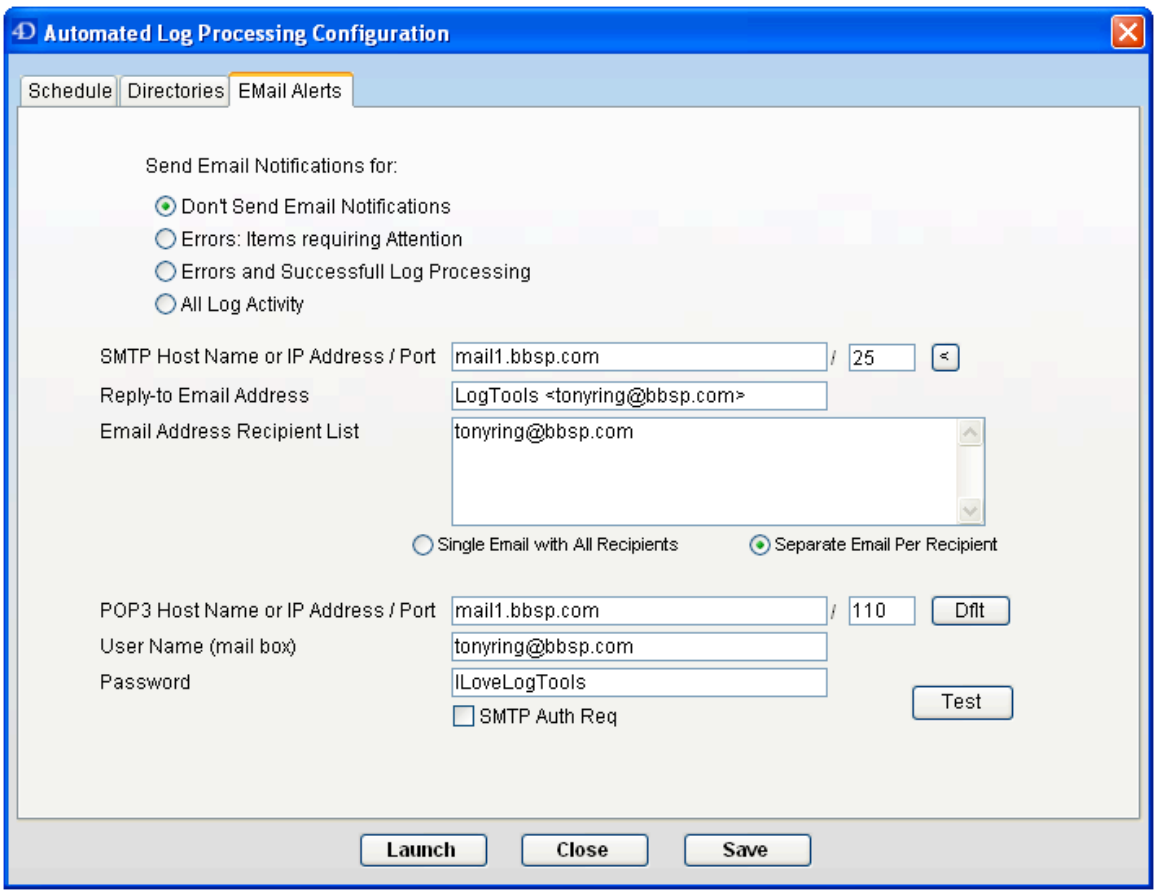
When using the 'Scan a Live log file' option, LogTools does NOT ever move or modify the live log file, or any of the backup log files.

Scan completed log files:
The 'Scan completed log files' option is for scanning log files that have been completed and closed by the database engine.

After LogTools scans the log file, LogTools will then move the log file to the directory of processed log files that you specify (in the directories tab).

Other than moving the processed log file, LogTools does not modify the log file.





Most of the information for the setup of Automated Log Processing is self-explanatory. Items requiring more explanation have a Help button adjacent to them to give you more information about it's function.

WINDOWS Menu

The Windows menu allows you to navigate between open windows in LogTools

NEW FEATURES IN VERSION 3.0

Here is a summary of what's new in LogTools version 3.0.

Automated Log Processing (an entire new module)

Ability to filter by Log Sequence numbers

LogTools is now written in 4D 2004 (was in version 6.8)

New, more colorful interface

When you, the user adjust column widths in any of the viewer windows (Log View, Table View and Record View) LogTools will now remember your adjusted width. This information is stored persistently in the LogTools data file.

NEW FEATURES IN VERSION 4.0

Here is a summary of what's new in LogTools version 4.0

4D Version 11 Log Compatibility
Users View window
Context Sensitive drop-down menus
Automated Log Processing: Scanning Live Log files
Improved export formats
Better support for viewing subtable information (via double-click)
Enable user to define Table names, Field Names and User Names on the fly

NEW FEATURES IN VERSION 4.1

Context View Window

NEW FEATURES IN VERSION 5.0

LogTools is now a V11 native-written application
Can display lists of over 100,000 items without crashing upon closing
Bugfixes with decoding Blob information
Support for V12 blobs stored in records
Support for parsing logs greater than 2GB
Assed a "Size" column to the Log view window, and to the Filtering window
Improved the Schemas Importing/Management dialog
We now provide v11 & v12 components to export a Schema that LogTools can consume