



# Agenda

- Who needs a log analysis tool?
- What is the IBM DB2 Log Analysis Tool?
- Robust data change reporting
- Rapid data restore/change reversal
- Enhancements in Version 1.3
- Overview
  - ▶ Log backward vs. log forward processing
  - ▶ Normal vs. continuous mode processing
  - ▶ Advanced BSDS Options
  - ▶ "Walk-through" of some panels to show navigation
- Summary

## Who needs a log analysis tool?

- DBAs and Application Programmers
  - responsible for application integrity
  - responsible for maintaining test environments
  - test new program functions with SQL
  
- Auditors
  - need to know who updated DB2 data
  - need to monitor sensitive databases and tables for unauthorized/inaccurate changes

## Slide - Who needs a log analysis tool?

The primary DB2 Log Analysis Tool users are DBAs and Application Programmers who are responsible for application integrity.

With advances in hardware technology, fewer hardware errors occur than before. More data is corrupted now from application errors. Recovery from an application problem is becoming a critical issue in many environments.

The DB2 Log Analysis Tool is able to move changes quickly between production and development systems and it allows you to isolate a specific set of changes to move. This feature assists development personnel to maintain development databases more efficiently. The tool is also used within programs when testing SQL to ensure that SQL is actually performing the updates you need.

Other DB2 Log Analysis Tool users are auditors. Auditors require such information as who updated data and who accessed a specific table. The tool enables them to monitor sensitive databases for unauthorized or inaccurate changes.

## What is the IBM DB2 Log Analysis Tool?

- A data “inspector”
- A data “corrector”
- Specializes in data changes
  - identifies
  - reports
  - audits
  - reverses
  - rolls data forward or backward to a specific point in time
  - applies changes from one database to another database

## Slide - What is the IBM DB2 Log Analysis Tool?

The DB2 Log Analysis Tool is a data inspector. It reviews the log and identifies what changes were made and by whom.

It is also a data corrector. Once it is determined what changes are in the log, it is possible to create reversal SQL that can either undo the SQL in the log or reapply the changes by creating "redo SQL" to apply those changes.

In summary, the DB2 Log Analysis Tool is used to identify log updates, create update reports, audit the log, reverse data changes, roll the data forward or go backward to a selected point in time, and move data changes from one database to another. Because the SQL generated by the tool is standard SQL, it can run on the system where it was generated or another DB2 in the same complex.

## Robust data change reporting

- Quickly distinguishes between
  - changes made by users via SQL
  - changes resulting from referential integrity constraints
- Filter reports by multiple criteria, including the following:
  - date/time
  - type of change (INSERT, UPDATE, etc.)
  - database, table, tablespace, column data
  - authid
- Summary reports
- Detail reports
  - reports on dropped objects
  - optional: you can load information about changed data to a DB2 table for advanced analysis

## Slide - Robust data change reporting (1 of 2)

With Data Reporting you can quickly identify SQL changes made by specific users. Updates to DB2 tables are always logged so you can review the log and extract those changes. For example, if auditors want to inspect what a particular user has done in DB2, they can identify all updates from that user quickly and easily.

Beginning with version 1.3, changes resulting from referential integrity constraint updates are flagged in the reports for ease of identification. Many filters are available to identify specific data. In addition to extracting information about a specific user, you can extract information such as data additions, changes, and deletions, or you can request information within a date and time range. You can limit the reports based on the database, tablespace, or an actual table column.

## Slide - Robust data change reporting (2 of 2)

You can filter on the AUTHID, other DB2 object identifiers, and additional types of information. Two types of reports are available:

- Summary Report - which summarizes the activity within the filter constraints
- Detail Report - which details each SQL updated in the log and the content of the row before and after the change.

The DB2 Log Analysis Tool also reports on dropped objects. Included with the tool is an interface to another IBM tool (DB2 Object Restore) which enhances reporting on dropped objects. You can load detail records into a DB2 table and use SQL to perform advanced analysis on the data you need to track.



## Rapid data restore/change reversal

- Quickly select undesired changes
- Automatically generate SQL
- Execute SQL to restore data to desired state
- Easy to select “point in time”
  - roll data backward/forward
  - select choices from ISPF panels
  - example: recover a dropped object from an old image copy and apply REDO SQL

## Slide - Rapid data restore/change reversal

An important feature of the Log Analysis Tool is the capability to remove unwanted changes to SQL.

For example, if someone updates credit card limits using the wrong multiplier, the Log Analysis Tool can identify all the updates to that database and generate SQL to reverse the changes.

If a tablespace becomes corrupted, and it still exists, and only a few changes must be removed, the tool can roll it backwards in time. Or, if you need to restore to an image copy from a prior point in time, the tool can create SQL to run forward from that point in time. You can also recover a dropped object from an old image copy and apply the SQL to bring it up to the point where it was dropped.

You use an ISPF panel interface to perform all these tasks.



## Rapid data restore/change reversal cont.

- Referential integrity support
- Create a static SQL program for applying large files of SQL
- Full support for ROWID type columns
- Use with DB2 Object Restore to bring objects back to original state even if they are no longer in the DB2 catalog



## Slide - Rapid data restore/change reversal cont.

The DB2 Log Analysis Tool provides referential integrity support which allows you to identify changes resulting from referential integrity restraints and not create the undo or redo SQL for the referential integrity updates, but only for the original updates.

A facility is also included that creates a static program to apply large SQL files which processes more efficiently than dynamic SQL.

The tool also fully supports ROWID type columns.

In addition to the report enhancing tool, DB2 Log Analysis Tool interfaces with a product called Object Restore. Object Restore can return objects to their original state even if they are no longer in the DB2 catalog. Upon restoring the data, the Log Analysis Tool then rolls the data forward to the current point of the drop.

## Enhancements in Version 1.3

- New rollback reporting
- Identify log records caused by DB2 referential integrity updates
- Enhanced dropped object reporting
- Column data filters specified in ISPF dialogs
- Advanced archive log specification
- Performance enhancements
- Specific partition processing
- Support for ROWID columns
- ASCII and Unicode support

## Slide - Enhancements in Version 1.3 (1 of 2)

Beginning with Version 1.3, Log Analysis Tool contains rollback reporting. Some installations frequently encounter application errors which inhibit completion of jobs. When an error occurs, it is necessary to do a rollback to return to a point before the error. This becomes expensive if done on a regular basis. With a rollback report, the DBA knows who caused the rollback and talks with the programmer to determine when the error occurred. By identifying the cause of the error, they can see if it is possible to revise the application to prevent future errors.

Also included in this release is a process to identify log records caused by DB2 referential integrity updates. This enables you to filter out the referential integrity updates when creating undo and redo SQL.

Some additional enhancements were added to the dropped object reporting feature. You can now report on those objects based on the specified new or old DB2 identifiers.

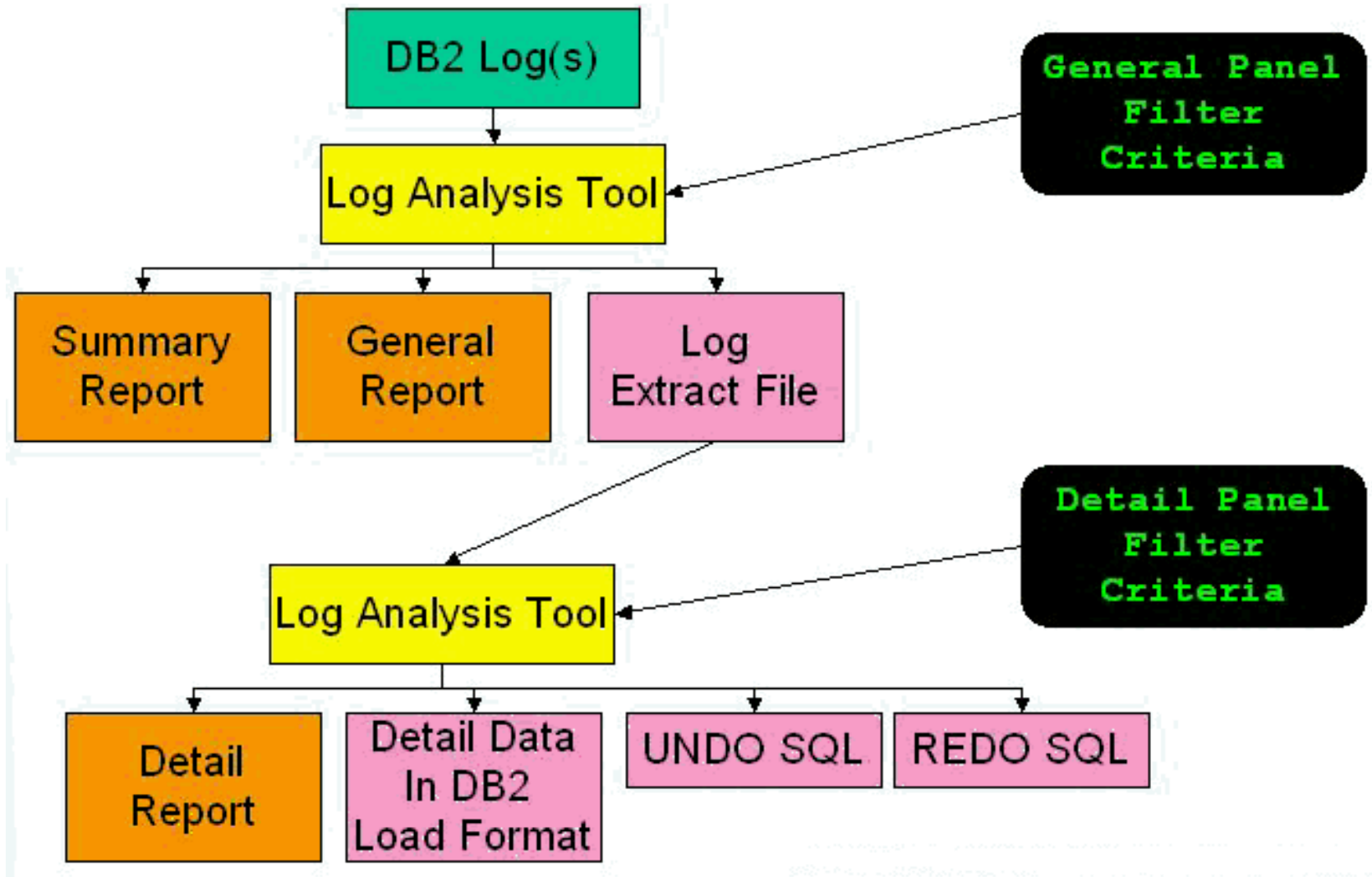
## Slide - Enhancements in Version 1.3 (2 of 2)

Column data filters are part of an ISPF dialog, making them easier to specify. In addition, there are advanced Archive Log specifications which provide control over which Archive Logs are used. In previous versions Log Analysis Tool read the bootstrap data set and used the primary copy in the Archive Log. Now, you can specify which Archive Log to use.

A number of performance enhancements were added in Version 1.3 making it run significantly faster than Version 1.2. Processing for specific partitions was also added allowing you to process individual partitions.

The ROWID column support is available beginning with Version 1.3 as well as enhanced ASCII support and support for Unicode data. If Log Analysis Tool encounters ASCII or Unicode data in a system 390 tablespace, it is translated to EBCDIC to make it readable in reports.

# Overview



## Slide - Overview

This schematic shows how Log Analysis Tool works. The input at the top of the screen is the DB2 log. On the General Report panel you specify the filter criteria, which are applied against the Log Analysis Tool to create a General Report. The General Report shows the units of recovery which qualify based on your filter criteria.

A Summary Report is produced which summarizes activity that matches the filters you specified. A Log Extract File is then created and provided as input to DTL processing. DTL processing generates the undo or redo SQL. At this point, you can change the filter criteria. For example, if you originally requested all updates for a table in a certain General Report, you can amend the filter for the Detail Report so that you receive only the changes for a particular column.

The Tool provides a Detail Report and a file you can load into DB2 and optionally undo or redo SQL. The undo SQL is used to roll backwards from the current system to a particular point in time. The redo is used to replay the SQL.

## Overview - Log backward vs. log forward processing

- Log backward processing
  - default operating mode
  - uses current DB2 data and works backward through the log to construct UNDO/REDO SQL
  
- Log forward processing
  - optional mode
  - starts with an IMAGE copy and processes forward from that point

## Slide - Log backward vs. log forward processing

Key processing modes:

- **Log Backwards:** The default operating mode. This mode begins on the DB2 object as it exists on DASD today and works backward from the log to build undo/redo SQL and detail images of the rows.
- **Log Forward:** An optional processing mode. This mode is required rows could have been moved as a result of running utilities against a tablespace. This is not detectable in Log Backwards mode.

## Overview - Normal vs. continuous mode processing

- Normal processing
  - considers only completed units-of-recovery within the time period specified
  
- Continuous mode processing
  - reports on all units-of-recovery completed within the time period specified
  - saves "in-flight" units-of-recovery for processing in the next run

## Slide - Normal vs. continuous mode processing

Processing begins with an image copy of the database and processes the log forward from the image copy to the end time specified in the Log Analysis Tool run.

You can also process normally in continuous mode. Normal processing includes only completed units of work within the time period specified in Log Analysis Tool. For example, if you specify that Log Analysis Tool should interrogate the log from 9:00 a.m. to 10:00 a.m. and you have a unit of work that started at 8:55 a.m. and ended at 9:05 a.m., that unit of work is not included in the reports. Only the units that started and ended within the specified time period are included.

In Continuous mode, you can start and run a process to the end of the log. Any units of recovery that are in-flight at the end of the process are saved in a file and are included in the next run. This way if you run Log Analysis Tool every hour, all changes are eventually reported.

## Overview - Advanced BSDS Options

- Use only ARCHLOG1
- Use only ARCHLOG2
- Active log priority
- Default for all options is “N”
- Explicitly specify log data sets

## Slide - Advanced BSDS Options

Beginning with Version 1.3, advanced bootstrap options are available. These options allow you to specify which archive copy of the log to use or to specify the active log as a priority. You normally would specify archive logs in order to prevent the Log Analysis Tool from interfering with DB2 processing.

The default for these options is "No" so that the new functionality is used. However, you can change the setting to "Yes" to use the old method.

There is also an option to explicitly specify the log data sets. If you want to use a particular data set, you can specify the data set name.



# Overview - Main Menu

```
DB2 LOG ANALYSIS TOOL FOR OS/390 -----
ENTER SELECTION ==> _

User ID: CSTHUB
Date...: 2003/01/06
Time...: 09:12
Release: 1.3.0

-----

1 Generate database activity report (general)
2 Generate database activity report (details)
3 Load detail data into DB2
4 Generate static SQL program
5 Execute dynamic SQL in batch
M Message Help
D Defaults
S Setup
T Tutorial
X Exit

-----

5655-E66 Copyright (C) 2002 IBM Corp. All rights reserved.
Copyright (C) 2002 Rocket Software, Inc. All rights reserved.
IBM is a registered trademark of International Business Machines Corporation.
Rocket is a registered trademark of Rocket Software, Inc.

-----
MA a 02/023
```

## Overview - Main Menu

The DB2 Log Analysis Tool Main Menu contains five options:

- Generate General Report
- Detail Report
- Load the Data Into DB2
- Generate A Static SQL Program
- Execute in Batch

This product does not contain many panels because it is primarily used to specify criteria to run a batch job.

Subsequent slides provide details on the underlying panels for each of those options.



# Overview - Generate general report

```

Generate database activity report (general) -----
COMMAND ==> _

DB2 subsystem name.. => I61A (SSID)   Action..... => E (E=Edit,S=Submit)
Generate details.... => N   (Y/N)     Job Identifier... =>
Data Sharing Mode... => Y   (Y/N)     Bypass SYSLGRNX.. => N (Y/N)
Specify Logs..... => N   (Y/N)

Log date/time range:
Start Date/Time..... => 2002/09/25 => 08:28:00 (YYYY/MM/DD HH:MM:SS)
End Date/Time..... => 2002/09/25 => 08:40:00 (YYYY/MM/DD HH:MM:SS)
Continuous mode file =>

Filters for log data:
Show UPDATES..... => Y   (Y/N)     Show INSERTs..... => Y (Y/N)
Show DELETES..... => Y   (Y/N)     Show Rollbacks... => N (Y/N/O)
Use object IDs..... => N   (Y/N)     Use object list.. => N (Y/N)
Table Owner..... =>
Database Name..... =>
Partition..... => 0   (0-254) AuthID..... =>
Job Name..... => CSTHUBA2 Plan Name..... =>
Catalog data..... => N   (Y/N)     Additional lists. => N (Y/N)

```

## Slide - Generate general report

When generating a general report, you can specify which log data set to use by entering "Y" for the Specify Log option.

The Use Object List option, the third item in the right column under *Filters for log data*, allows you to specify a list of objects.

The Additional Lists option, the last item under *Filters for log data*, provides the ability to specify additional filtering criteria.

These two options were added in Version 1.3.



# Overview - Specify lists of objects

```
Object List Filter -----  
COMMAND ===>
```

Hit 'Enter' to use the list; 'PF3' to bypass using the list

Object Type(T/S)	Table Creator/ Database Name	Table Name/ Tablespace Name
T	IBM151	RESTAURANTS
S	IBMDB151	STATES

## Slide - Specify lists of objects

The Object List filter screen enables you to specify tablespace or table filters.

For example, the first line of the data on the sample screen is a table filter for the table owner of IBM151 and table name of Restaurants. The second line is a tablespace filter which gives you the database name and tablespace name.

If you press PF3 after specifying the filter criteria, the filter data is ignored. Using PF3 provides the opportunity to change the filters, even while you are specifying them. If you press ENTER, the filters are used.



# Overview - Specify additional list filters

```
Additional Lists -----  
COMMAND ===>
```

Hit 'Enter' to use the list; 'PF3' to bypass using the list

Filter Type (A/J/P)	Authid/Job Name/ Plan Name
A	IBMUSER
P	IBMPPLAN
J	IBMJOB1

## Slide - Specify additional list filters

The Additional List sub-screen enables you to enter the Authid, plan filters, and the job name. As shown in the sample screen, they are specified by the single character filter type followed by the name of the filter.

Wildcards are also supported; so, for example, you can enter *IBM%* to filter any item beginning with IBM.

## Overview - Generate detail report

```

Generate database activity report (details) -----
COMMAND ==> _

Action..... ==> E           (E=Edit,S=Submit)
Job Identifier.... ==>           (8 character job identifier)
Generate Undo SQL. ==> Y       (Y/N)
Generate Redo SQL. ==> N       (Y/N)
Commit scope..... ==> 001     (000-999)
Apply log forward. ==> N       (Y/N)

Filters for the recovery file:
All in file..... ==> N (Y/N)   Show UPDATEs..... ==> Y (Y/N)
Show INSERTs..... ==> Y (Y/N)  Show DELETEs..... ==> Y (Y/N)
Table Owner..... ==>           Table Name..... ==>
Database Name.... ==>           Tablespace Name.. ==>
Job Name..... ==> CSTHUBA2     Plan Name..... ==>
AuthID..... ==>               URID..... ==>
Column Data..... ==> N (Y/N)

```

## Slide - Generate detail report

The Detail Report screen enables you to select the information to include in the Detail report.



## Overview - Load data into DB2

```

Load detail data into DB2 -----
COMMAND ==>

Action..... ==> E          (E=Edit,S=Submit)
Job Identifier.... ==>      (8 character job identifier)
Header..... ==> Y (Y/N)    Change Flag..... ==> Y (Y/N)
Data Only..... ==> N (Y/N) Create target tables ==> N (Y/N)
Database Name.... ==>      Tablespace Name..... ==>
Log..... ==> Y (Y/N)
LOAD specification ==> 0   (0-ALL, 1-I, 2-D, 4-UPOST, 8-UPRE)

Source fields:                Target fields:
SSID..... ==> I61A           SSID..... ==> I61A
Table Owner/Name (LOAD from): Table Owner/Name (LOAD to):
    /                          /
    /                          /
    /                          /
    /                          /
    /                          /
    /                          /
    /                          /
    /                          /
    /                          /
    /                          /
  
```

## Slide - Load data into DB2

The Load Data Into DB2 screen enables you to map source fields to target fields in the load process.

The change flag indicator, located on the third line of the second column, sets a flag for changed columns regardless of the value. An example of when setting the flag to Yes is helpful would be when an auditor needs to know whether any values in a specific table column were changed.

# Generate static SQL program

```
Generate static SQL program -----  
COMMAND ==>  
  
Action..... ==> E          (E=Edit,S=Submit)  
Job Identifier..... ==>          (8 character maximum)  
  
DB2 subsystem name.. ==> I61A      (DB2 SSID)  
DB2 Version..... ==>          (5.1, 6.1, or 7.1 only)  
DB2 macro library... ==> DSN.V610.SDSNMACS  
User plan name..... ==> LATSAMP1  
User DBRM library... ==> CSHUB.DBRMLIB  
User load library... ==> CSHUB.LOADLIB  
User source library. ==> CSHUB.SOURCE.ASM  
User program name... ==> LATSAMP1  
Commit Scope..... ==> 001        (000-999)  
Column Data..... ==> N          (Y/N)  
  
Use below fields to include a single table only (optional)  
Table Owner..... ==>  
Table Name..... ==>
```

## Slide - Generate static SQL program

The Generate Static SQL program screen enables you to set the criteria for the SQL program you are generating. A high-speed batch SQL processor enhancement was added in Version 1.3. The enhancement eliminates the need to generate the static SQL.

To set the criteria in this screen it is necessary to know DB2 libraries, subsystem names, and various library names.

# Overview - Program to process dynamic SQL in batch

```
Execute dynamic SQL in batch -----  
COMMAND ==>  
  
Action..... ==> E          (E=Edit,S=Submit)  
Job Identifier..... ==>          (8 character maximum)  
  
DB2 subsystem name.. ==> I61A   (DB2 SSID)  
Restart JCL..... ==> N         (Y/N)  
  
Use below fields to include a single table only (optional)  
Table Owner..... ==>  
Table Name..... ==>
```

## Slide - Program to process dynamic SQL in batch

Generating a batch program to process the SQL dynamically is more efficient than processing regular dynamic SQL, and entering specifications is simpler, as well.

# Tutorial

```
DB2 Log Analysis Tool - General usage tutorial ----- TUTORIAL  
COMMAND ==> _
```

Enter a selection number below, or use these keys to navigate:  
PF10 - prev page; PF11/Enter - next page; PF7 - back to this page.

- 0 General overview on product usage
- 1 Generate database activity report (general)
- 2 Create VSAM file for database activity report (details)
- 3 Generate the database activity report (details)
- 4 Load detail data into DB2
- 5 Report description (general)
- 6 Report description (summary)
- 7 Report description (details)
- 8 Messages generated during report generation
- 9 Generation of UNDO/REDO SQL
- 10 Log-backward versus log-forward processes
- 11 Continuous mode process
- 12 Processing the generated REDO/UNDO SQL (dynamic/static SQL)
- 13 Using dropped-object mode
- 14 Handling non-EBCDIC DB2 data (i.e., ASCII or UNICODE)
- 15 Miscellaneous notes, tips, performance, JCL restart information

## Slide - Tutorial

The tutorial for Log Analysis Tool was moved to the main panel in Version 1.3 so that you can locate it more easily than in previous versions.

The tutorial is very helpful for learning about the tool.



## Summary

- Application level recovery
  - UNDO/REDO SQL
  
- Auditing
  - Who did what when
  
- Continuous mode processing