

# ASG-Tracer<sup>®</sup> User's Guide

Version 7.4

Publication Number: TCR0200-74

Publication Date: February 2003

The information contained herein is the confidential and proprietary information of Allen Systems Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Allen Systems Group, Inc.

© 2003 Allen Systems Group, Inc. All rights reserved.

All names and products contained herein are the trademarks or registered trademarks of their respective holders.



---

# Contents

---

<b>Preface</b> .....	v
<b>About this Publication</b> .....	v
<b>ASG-Replication Suite Products</b> .....	vi
<b>Related Products</b> .....	vi
<b>Publication Conventions</b> .....	viii
<b>ASG Customer Support</b> .....	viii
<b>ASG Documentation/Product Enhancements</b> .....	xi
<b>1 Introduction to Tracer</b> .....	1
<b>Introduction</b> .....	1
<b>Tracer Features</b> .....	2
<b>Tracer Functions</b> .....	3
<b>How Tracer Works with the IDMS Journal</b> .....	5
<b>Tracer Components</b> .....	6
<b>2 Tracer Installation</b> .....	11
<b>ASG Service Pack</b> .....	11
<b>Technical Requirements</b> .....	12
<b>Installing Tracer</b> .....	12
<b>Tracer Naming Conventions</b> .....	13
<b>Setup Steps</b> .....	14
<b>3 Getting Started with Tracer</b> .....	19
<b>Step 1 - Create Dbkey of Relocated Records</b> .....	21
<b>Step 2 - DMCL and Schema Mapping</b> .....	21

Step 3 - Journal Reformat and Reports Job .....	25
Step 4 - Remove Rolled-back Run Units Job .....	27
Step 5 - Read Reformatted File and Reports Job.....	29
Step 6 - Run Unit Statistics Job .....	29
<b>4 Specialized and Advanced Topics .....</b>	<b>31</b>
Processing Non-quiesced Journal Archive Tapes .....	31
Splitting Reformatted Journal Records .....	34
Incremental Refreshing of Parallel Databases.....	35
<b>5 Control Parameters .....</b>	<b>37</b>
Run Control Parameters .....	37
Report Control Parameters .....	39
Default Input File (Tracer) Parameters .....	72
Field-level Selection Parameters .....	73
SQL Selection Parameters .....	76
<b>6 Tracer Report Facility.....</b>	<b>101</b>
Tracker (Field-level) Report .....	101
Record-level (Reformatter) Reports .....	108
Database (Tuning and Statistics) Reports.....	130
<b>Appendix A</b>	
<b>Reformatted File Record Layout .....</b>	<b>161</b>
Reformatted File Record Occurrence Layout .....	161
DBTRFMTJ File Fields (SEL-YN-DBT-WRITE-REFORMATTED=Y).....	168
Source Journal Fields .....	168
Load Module Fields (from DBTRACE) .....	170
<b>Appendix B</b>	
<b>Parameters Affecting Relocated Records.....</b>	<b>185</b>
<b>Appendix C</b>	

**CA-IDMS System Exit 14 Advanced Customization Techniques** ..... 189

**Customization Procedures** ..... 189

**Appendix D**

**Creating a Dbkey of Relocated Records** ..... 197

    Run Job @RAALOOK..... 197

    Run Job @RABJCLB..... 197

    Run Job @RA0BSWE ..... 201

    Run Job @RB0BSOR..... 201

    Run Job @RC0BCTL..... 201

    Run Job @RD0BVSM ..... 202

    Run Job @RE0BLOD..... 202

**Appendix E**

**Frequently Asked Questions** ..... 203

**Index**..... 209



---

## Preface

---

This *ASG-Tracer User's Guide* provides a detailed explanation of how to install, invoke, and use the ASG-Tracer (herein called Tracer) product.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on the Tracer product.

References to CA-IDMS, except for specific releases and product names will be IDMS.

## About this Publication

This *ASG-Tracer User's Guide* consists of these chapters:

- [Chapter 1, "Introduction to Tracer."](#) describes the Tracer product and how it can be used at your site.
- [Chapter 2, "Tracer Installation."](#) describes the steps required to install Tracer. These steps also cover installation of the optional CA-IDMS System Exit 14.
- [Chapter 3, "Getting Started with Tracer."](#) explains how to implement the Reformat, Rollback, and Tracker functions of Tracer.
- [Chapter 4, "Specialized and Advanced Topics."](#) explains how to process non-quieted journal archive tapes, use the work file, split reformatted journal records, and perform an incremental refresh of parallel databases.
- [Chapter 5, "Control Parameters."](#) explains the various parameters and selection criteria used by Tracer.
- [Chapter 6, "Tracer Report Facility."](#) describes the Tracer reports and explains how to implement each one.

## ASG-Replication Suite Products

The ASG-Replication Suite consists of these products:

ASG Product	Herein Called
ASG-Replication Suite	Replication Suite
ASG-Replication Agent for Real-Time	Replication Agent for Real-Time
ASG-Replication Agent	Replication Agent
ASG-Table Designer	Table Designer
ASG-Journal Manager	Journal Manager
ASG-Tracer	Tracer
ASG-Rascle	Rascle
ASG-SQL Optimizer	SQL Optimizer

---

## Related Products

This table lists the related products for Replication Suite:

ASG Product	Herein Called
ASG-MQ Database Envoy	Database Envoy
ASG-MQ Envoy for DB2	MQ Envoy for DB2
ASG-Outbound Enterprise	Outbound Enterprise
ASG-Outbound Express	Outbound Express

---

**Note:** \_\_\_\_\_

For more information about any of these products, contact your ASG sales representative.

---

---

## Related Publications

### ASG Publications

The complete documentation library for Replication Suite consists of these publications (where *nn* represents the product version number):

- *ASG-Journal Manager User's Guide* (JMI0200-*nn*) provides an overview of ASG-Journal Manager's operation, step-by-step procedures for installation and configuration, and instructions for using and maintaining the product.
- *ASG-Rasclé User's Guide* (RAS0200-*nn*) provides a detailed explanation of how to install, customize, and run the ASG-Rasclé product.
- *ASG-Replication Agent User's Guide* (REA0200-*nn*) provides an overview of ASG-Replication Agent's operation, step-by-step procedures for installation and configuration, and instructions for using and maintaining the product.
- *ASG-Table Designer User's Guide* (TAD0200-*nn*) provides step-by-step procedures for installation and use of ASG-Table Designer.
- *ASG-Tracer User's Guide* (TCR0200-*nn*) provides a detailed explanation of how to install, invoke, and use the ASG-Tracer product.
- *ASG-Replication Suite Real-Time Option User's Guide* (ROO0200-*nn*) provides detailed information about ASG-Replication Suite Real-Time Option (herein called *Replication Agent Suite Real-Time Option*).
- *ASG-SQL Optimizer User's Guide* (SON0200-*nn*) discusses the benefits of ASG-SQL Optimizer and explains how SQL Optimizer works to reduce processing time required for replicating database updates.

**Note:** \_\_\_\_\_

To obtain a specific version of a publication, contact ASG Customer Support.

---

### Sybase Corporation Publications

ASG recommends that you become familiar with the Sybase Open ClientConnect and DirectConnect publications. You can request these publications from the Sybase Corporation, or you can view them online at <http://www.sybase.com>, the Internet Sybooks website.

### IBM Corporation Publications

ASG also recommends that you become familiar with the IBM Corporation MQSeries publications. You can request these publications from the IBM Corporation, or you can view them online at <http://weplib.mainz.ibm.com>.

## Publication Conventions

ASG uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
<i>lowercase italic monospace</i>	Information that you provide according to your particular situation. For example, you would replace <i>filename</i> with the actual name of the file.
Monospace	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.
Vertical Separator Bar ( ) with underline	Options available with the default value underlined (e.g., Y  <u>N</u> ).
<u>Underline</u>	Denotes a cursor-selectable field or line.

## ASG Customer Support

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours.

**ASG Third-party Support.** ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

## Intelligent Support Portal (ISP)

Online product support is available at: <http://www.asg.com/support/support.asp> via the ASG Intelligent Support Portal (ISP). Your logon information for ISP online support is:

Customer ID = *NNNNNNNNNN*

Password = *XXXXXXXXXX*

where:

*NNNNNNNNNN* is your customer ID supplied by ASG Product Distribution.

*XXXXXXXXXX* is your unique password supplied by ASG Product Distribution.

The *ASG-Intelligent Support Portal User's Guide* provides instructions on how to use the ISP and is located on the ASG Support web page.

## Telephone Support

To expedite response time, please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely as displayed
- A description of the specific steps that immediately preceded the problem
- Verify whether you received an ASG Service Pack or cumulative service tape for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack. You can access the latest software corrections and Service Packs via the ISP.
- The severity code (ASG Customer Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)

### Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

### *The Americas*

	<b>Phone</b>	<b>Fax</b>	<b>E-mail</b>
<b>United States and Canada</b>	800.354.3578	1.703.464.4901	support@asg.com

### *Europe, Middle East, and Africa (EMEA)*

During normal business hours, we recommend that you call the Central Support number first (except in South Africa).

	<b>Phone</b>	<b>Fax</b>	<b>E-mail</b>
<b>Central Support</b>	00.800.3544.3578	44.1727.812018	support.emea@asg.com
<b>English</b>	44.1727.736305	44.1727.812018	support.uk@asg.com
<b>French</b>	33.141.028590	33.141.028589	support.fr@asg.com
<b>German</b>	49.89.45716.200	49.89.45716.400	support.de@asg.com
<b>Italian</b>	39.0290450025		support.it@asg.com
<b>Dutch</b>	31.30.241.6133		support.nl@asg.com
<b>Spanish</b>	34.913.523.800	34.917.156.961	support.es@asg.com
<b>South Africa</b>	800.201.423		support.sa@asg.com

### *Asia Pacific (APAC)*

	<b>Phone</b>	<b>Fax</b>	<b>E-mail</b>
<b>Central Support</b>	61.3.9645.8500	61.3.9645.8077	support.au@asg.com
<b>Australia</b>	800.637.947	61.3.9645.8077	support.au@asg.com
<b>Hong Kong</b>	800.96.2800		support.hk@asg.com
<b>Japan</b>	81.3.5326.3684	81.3.5326.3001	support.au@asg.com
<b>Singapore</b>	65.224.3080	65.224.8516	support.sg@asg.com

### *All Other Countries (Also for any non-working numbers)*

	<b>Phone</b>	<b>Fax</b>	<b>E-mail</b>
<b>All other countries</b>	1.239.435.2201		support@asg.com

If you receive a voice mail message, follow the instructions to report a production-down or critical problem. Leave a detailed message including your name and phone number. An ASG Customer Support representative will be paged and will return your call as soon as possible. Please have available the information described previously when the ASG Customer Support representative contacts you.

## **ASG Documentation/Product Enhancements**

Submit all product and documentation suggestions to ASG's product management team at <http://www.asg.com/asp/emailproductsuggestions.asp>.

If you do not have access to the web, FAX your suggestions to product management at (239) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.



---

# 1

## Introduction to Tracer

---

This chapter provides an introduction to Tracer and contains these sections:

Section	Page
<a href="#">Introduction</a>	<a href="#">1</a>
<a href="#">Tracer Features</a>	<a href="#">2</a>
<a href="#">Tracer Functions</a>	<a href="#">3</a>
<a href="#">How Tracer Works with the IDMS Journal</a>	<a href="#">5</a>
<a href="#">Tracer Components</a>	<a href="#">6</a>

---

### Introduction

Tracer reformats database record images from the IDMS journal archive files, converting them from a cryptic state to the format that was used in the original updating application program. With Tracer, you can utilize the valuable information that IDMS automatically writes to its journal files about your database updates.

Tracer reformats the data images of updated record occurrences. These images are then available to you exactly as they were in the application program's original image—immediately before and after each update. Prior to Tracer, user-written application programs could not practically access the journaled before/after database images.

In addition to reformatting records, Tracer builds files containing records that detail end-run-unit-ended (ENDJ) or abort (ABRT) check points, archived records, and end-of-job statistics. You can use these files as input to your own applications and reports. Or, you can use the pre-written programs included with Tracer to produce standard reports about database update activity. In addition, the Tracer module can perform a rollback operation to exclude update images from the files that IDMS rolls out either when an abort occurs or when an applications program uses the rollback DML verb.

## Tracer Features

Tracer provides these features to assist with database updates:

**Reformats IDMS journaled records.** Tracer can reformat any journaled images of updated IDMS record occurrences. For example, Tracer converts journaled IDMS before and after record images to the format that was used in the program that originally issued the database update verb.

**Writes USER information to the IDMS journal archive.** An optional BIND/READY Exit (IDMS System Exit 14) comes with Tracer. Through this exit, Tracer writes a user journal record to the IDMS journal archive file. This USER record contains the user ID, terminal ID, task name, and updating subschema associated with each updating network run unit.

**Creates a reformatted journal archive file.** You can configure Tracer to a reformatted journal archive file that contains only journaled information matching the selection criteria Tracer provides. This file contains the information produced by the System Exit 14. It also provides run unit information, which includes the program name and the record image as it displayed to the updating application program.

**Produces audit reports.** The reformatted journal archive file enables a site to trace all updates, thus providing audit trail functionality for any existing IDMS update programs. When a MODIFY IDMS DML verb is responsible for the journaled images, Tracer places both the before and after image of the updated database record occurrence into a single record in the reformatted file. This audit trail feature facilitates analysis of changes made to specific fields within each updated record occurrence.

**Produces files for defining user-based reports.** Tracer produces a file containing information from ABRT and ENDJ records that were written to the journal file. This file contains information showing how ENDJ information is used with the reformatted file to exclude the updates that IDMS rolled back. A stamp file, //DBSTAMP, contains information that is output when the IDMS System User Exit 14 is used. End-of-job information and statistics are output in a run unit file, //DBTRUNUI. User-defined reports utilize these files to report on site-specific needs.

**Reports individual updated record elements.** Tracer produces a report displaying the before and/or after images of each individual record element (field) in any IDMS record occurrence that was stored, erased, or modified.

## Tracer Functions

Tracer provides many functions for database analysts, systems technicians, and application programmers. You can effectively use Tracer to perform these tasks:

**To Trace modifications to any data.** Using selection criteria, your site can specify which sensitive data items Tracer monitors. Tracer produces reports that answer the following questions about all modifications to sensitive data:

- What data element was updated?
- Which user ID caused the update?
- What was the data value of the field(s) before the update?
- What is the data value of the field(s) after the update?
- Which program and subschema were used to update the data?
- Which terminal or batch job was used to update the data?
- Which user ID initiated the update?

Without Tracer, IDMS sites must develop audit trail routines within application programs to store audit database records each time the program changes sensitive data. Tracer relies only on the information automatically written to the journal by IDMS. This allows sites to trace database updates performed by your application programs without requiring coding changes.

Application programs no longer need to store special audit records. There is no need for additional physical page writes to the IDMS database and the journal files. This significantly improves online performance throughput.

**To provide the framework for incremental refreshes of non-IDMS parallel files.** The reformatted journal archive file that Tracer produces may be used to help synchronize non-IDMS files with IDMS master database counterparts. User-written application programs may use this reformatted file to apply changes to the target database that were made to the master IDMS database since the last incremental refresh.

This process is often preferable to the total rebuild approach. It is a more efficient way to keep relational databases parallel with their IDMS master database counterparts, whether they are on PC LANs, DEC machines, or IBM/DB2 environments.

**Analyzes the performance of long-running update run units.** Tracer can help your site determine why run units are taking too much time—whether it is due to excessive updating of pages by the DBMS system overhead, or whether it waits on record occurrences locked by concurrent update run units. By reporting even the system type updates, Tracer can help reduce system overhead by informing DBAs and programmers of needed changes to the physical database design or application program updates.

**To correct damage from program errors.** Tracer may be used to create a comprehensive, easy to use view of what happened to records that were updated incorrectly by a program error, or minor update bugs, not discovered until after the damage is done. If the problem was not too severe, a site-coded fix program may be able to use the Tracer reformatted file to determine which record occurrences were incorrectly updated. It can then return only the damaged elements to their original value.

**To recover from catastrophes.** Tracer provides a form of insurance in case a site cannot completely recover from database corruption using the IDMS roll back or roll forward utilities. Tracer produces a report of data that must be re-entered after the database is restored and rolled forward to the most recent point. When you install the IDMS user exit (EXIT-14), this report can then be sorted by the user ID, program name, and task ID.

Tracer reports can help you recover when a program containing a bug corrupts the database. Often, the only remedy is to restore the database from the last backup and rerun the fixed batch program. Online transactions entered after the last backup must also be re-entered (i.e., the manual recovery period).

Typically, there is no hard copy list that itemizes changes made during the manual recovery period. Tracer enables your site to use the journal archive tapes to quickly produce a listing of each user's changes. This list shows which records and record elements were modified by users during the manual recovery period.

In summary, Tracer provides solutions through several COBOL programs—complete with source code. These programs provide useful reports that require no further programming. They also provide an example to help code site-specific applications using the Tracer reformatted journal.

Tracer also serves as a software toolkit for extracting information from the IDMS journals about how the database is changed. It simplifies the development of site-specific applications that are only useful after Tracer reformats the journaled database record images. Your site can produce reports with a few lines of code via COBOL, Easytrieve Plus, CULPRIT, or ASG's own fourth-generation language product, ASG-GUEST-IDT.

## How Tracer Works with the IDMS Journal

IDMS writes to its journal files a snapshot image of each database record as it looked immediately preceding (BEFORE image) and following (AFTER image) an update. The program writes one or more journal entries for every DML update verb issued by any program updating a database running under the IDMS Central Version (CV). This includes all online update programs and any batch local mode programs that write to a journal file.

The IDMS DBMS software writes the before and after database change images to provide such features as:

- In Flight Automatic Rollback of abending transactions
- Roll Forward Processing

You can use these features to recover from a DASD head crash or for other hardware problems where a IDMS database resides.

Tracer uses the information from the IDMS journal archive file in a manner not provided for by standard IDMS software. It reformats the before/after images from a cryptic compressed, spanned, and fragmented condition into the format that was used in the original application program.

Tracer uses the following information as input to each Tracer run:

- Information extracted from the directory
- Output of the IDMSRPTS utility program

Tracer then uses the information to decide when and how to despan, defragment, and decompress the IDMS journal images.

Tracer contains COBOL programs that call Assembler programs to do the reformatting and, optionally, produce the reformatted file. (See ["Reformatted File Record Layout" on page 161](#) for a description of the Tracer COBOL record layouts and ["Control Parameters" on page 37](#) for a description of selection criteria fields.)

You can use Tracer to:

- Produce standard reports from the IDMS journal archive file
- Create customized reports using the Tracer-produced output files
- Create your own output files to further adapt functions to suit your site's requirements.

## Tracer Components

Tracer provides a variety of programs and reports to help you use the contents of the IDMS journal archive file with virtually no programming effort. Usually, only minor parameter setting adjustments in any of the pre-written reports are necessary.

Tracer consists of two basic components:

- Reformat facility
- Reporting facility
  - Record level reports
  - Field-level reports (Tracker)
  - DBA statistics reports

### *Reformat Facility*

The Reformat facility reads the journal archive files, which are produced by the journal archive process, and produces the Tracer reformatted journal file (DBTRFMT).

### *Reporting Facility*

From the data obtained by the Reformat facility, you may produce record level reports or database statistics and/or performance reports. You may run the DBTRFMT Reformat program, against a IDMS journal archive file to produce these reports:

- record level reports
- database application reports
- field-level reports

### *Record-Level (Reformatter) Reports*

These are Tracer's record-level (reformatter) reports:

**Report 01.** This report tracks, at the record occurrence level, all physical record updates that IDMS performs for all programs.

**Report 02.** This report tracks all updates, by user, made with the ASG-DBOL or DMLO programs for any subschema.

**Report 05.** This report provides more detailed information collected from the Connect and/or Disconnect verb.

**Report 11.** This report tracks all physical IDMS updates. This report may be used to display the full hexadecimal view of all data affected by any update operation.

### Field-Level Reports (Tracker)

This function produces a comparison report on record and field attributes. The Tracker facility procedures used to produce this report include:

- Schema Navigation
- Schema Mapping
- Record Comparison.

### Tracker Facility

Tracker uses a schema navigation program to produce a non-executable load module containing the record and field definitions. You need to keep the load module to run against old journal files, even after data on the dictionary and the current database has been restructured.

### Tracker's SQL-like Language Selection Criteria

With SQL-like selection criteria, you can limit which records are printed. This is an example:

```
//FLDSELS DD *  
SELECT ESALHIST-8002 WHERE DELTA% SALARY-AMT > 10 OR DELTA  
SALARY-AMT > 450.00
```

These are ways to use SQL-like field-level selection:

- List only salary records where the new salary amount is more than 10% (DELTA%) greater than the old salary amount.
- List only changes made to a specific department or part number.
- List only changes made to EMPLOYEE records where EMPLOYEE-NAME contains the value SMITH.
- In order to determine who deleted a record and what information the record contained (so you could re-enter it using DBOL or DMLO). Use field-level selection criteria to select the CALC-KEY of the before record image.

SQL-like selection criteria also provides the ability to specify (limit) the names of records and elements which you will see on the Tracker report as shown here:

**Figure 1 • Limiting Tracker Report Data**

```
//FLDINCL DD *  
EXC ALL  
INC SR-036  
EXC SR-036 DATE-LU-036  
EXC SR-036 PREP-BY-036  
INC INQ-058
```

Another feature is the ability to print the Tracker report in 80-column width. When this feature is enabled, the before/after images print in a list, which makes the report longer, but easier to read on a terminal.

### *DBA Statistics Reports*

The DBA statistics reports provide tuning data and statistics on run units, such as:

- Concurrent run units
- Run unit resources
- In association with record locks

These are the statistics reports (30 through 41):

**DBTRPT30.** Concurrent Run Units Detail Report

**DBTRPT31.** Concurrent Run Units with Suspected Record Locks Report

**DBTRPT33.** User Report by Application

**DBTRPT34.** Run Unit Reports

**DBTRPT37.** Concurrent Run Unit Summary Reports

**DBTRPT38.** Program Resource Reports (reports 39, 40, and, 41)

**DBTRPT39.** Program Resource Report by Program Name

**DBTRPT40.** Program Resource Report by Most Used

**DBTRPT41.** Program Resource Report by Longest Run

**DBTFIGHT.** Contention and Record Locking Report

**Tracer Summary.** Record Update Counts and Totals

## CA-IDMS System Exit 14

CA-IDMS System Exit 14 is optional. If installed at your site, you can use this exit to write additional information to the IDMS journal archive file for IDMS network database run units. Tracer invokes this exit each time a BIND RUN UNIT or a READY area occurs in the IDMS CV. The System Exit 14 allows Tracer to locate the following:

- User ID
- Terminal ID
- Task name
- Subschema name associated with the task
- Run unit responsible for each journaled record

## Schema Mapping

Schema mapping creates a load module that enables Tracer to interpret the journal records using accurate record layouts of the database records. This allows Tracer to access data at the field (record element) level for both reporting and SQL-like selection.

## Other Advanced User Topics

**Non-quieted journals.** When journal reports are dynamically created throughout the day, there is no quiesce point at the end of each journal tape. When Tracer encounters the end of the input journal archive file, it can optionally save information about each in-progress run unit (e.g., the sequence number of the most recent checkpoint) to the unfinished file and then use this information when beginning to read the next journal archive file. The DBTROLBK (rollback) utility also uses the unfinished file to remove aborted transactions.

**User-written custom journal reports.** You may create your own journal reports through a user-written COBOL sub-program that Tracer invokes directly as an Exit program.



---

# 2

## Tracer Installation

---

This chapter describes the steps required to install Tracer and contains these sections:

Section	Page
<a href="#">ASG Service Pack</a>	<a href="#">11</a>
<a href="#">Technical Requirements</a>	<a href="#">12</a>
<a href="#">Installing Tracer</a>	<a href="#">12</a>
<a href="#">Tracer Naming Conventions</a>	<a href="#">13</a>
<a href="#">Setup Steps</a>	<a href="#">14</a>

After installing the Tracer product, refer to ["Getting Started with Tracer" on page 19](#) and ["Control Parameters" on page 37](#), respectively, for operating and customization instructions.

### ASG Service Pack

Verify whether you received an ASG Service Pack for this product. If so, read the instructions for installing the Service Pack before proceeding with the product installation. The installation instructions are located in a text file on the distribution media included with the Service Pack. If you have any problems with the Service Pack, contact the ASG Customer Support.

## Technical Requirements

Tracer installs under MVS/XA or MVS/ESA. Tracer supports CA-IDMS Release 12.x, 14.x, and 15.x. You must install our User Exit 14 or modify your User Exit 14 if you want to capture subschema, user ID, and task information to be included on Tracer reports and output files.

Your installation tape includes several JCL and data files that you can use to test and verify your installation of Tracer.

## Installing Tracer

### Library Allocation

Use this table of attributes to allocate the libraries required by the Tracer distribution tape off-load and Tracer installation:

DSN	RECFM	LRECL	BLKSIZE	SPACE
TCR.Rxxx.LOADLIB	U	0	32760	CYL,(5,1,30)
TCR.Rxxx.JCLLIB	FB	80	22880	CYL,(2,1,30)

### Upload the Tracer Product Files

Create the JCL shown in [Figure 2](#) to upload the Tracer product files. Use the dataset names that were allocated in "[Library Allocation](#)" on page 12.

**Figure 2 • JCL to Upload Tracer Product Files**

```
//JOBNAME JOB (ACCT), 'TRACER INSTALL', NOTIFY=YOURUSERID,
// CLASS=YOURCLASS,MSGCLASS=YOURMSGCLASS
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//LOADIN DD DISP=OLD,DSN=TCR.Rxxx.LOADLIB,
// UNIT=TAPE,LABEL=(1,SL),VOL=(,RETAIN,,SER=TCRxxx)
//JCLIN DD DISP=OLD,DSN=TCR.Rxxx.JCLLIB,
// UNIT=TAPE,LABEL=(2,SL),VOL=(,RETAIN,,SER=TCRxxx)
//LOADOUT DD DSN=TCR.Rxxx.LOADLIB,DISP=SHR
//JCLOUT DD DSN=TCR.Rxxx.JCLLIB,DISP=SHR
//SYSIN DD *
COPY INDD=( (LOADIN,R) ),OUTDD=LOADOUT
COPY INDD=( (JCLIN,R) ),OUTDD=JCLOUT
```

## Tracer Naming Conventions

These are the naming conventions for all Tracer jobs:

- The first character is always the at sign (@), which sorts to the top of the PDS.
- The second character indicates the job category:

I = Install jobs  
R = Relocated record jobs  
S = Setup CV specific data sets  
T = Tracer jobs

- The third character indicates the sort sequence (a, b, c, etc.).
- The fourth character indicates the database release level:

0 = CA-IDMS Release 12.x, 14.x, and 15.x  
2 = CA-IDMS Release 12.x only  
3 = CA-IDMS Release 14.x only  
4 = CA-IDMS Release 15.x only

- The fifth character indicates the database type:

B = Both network and relational  
N = Network  
R = Relational

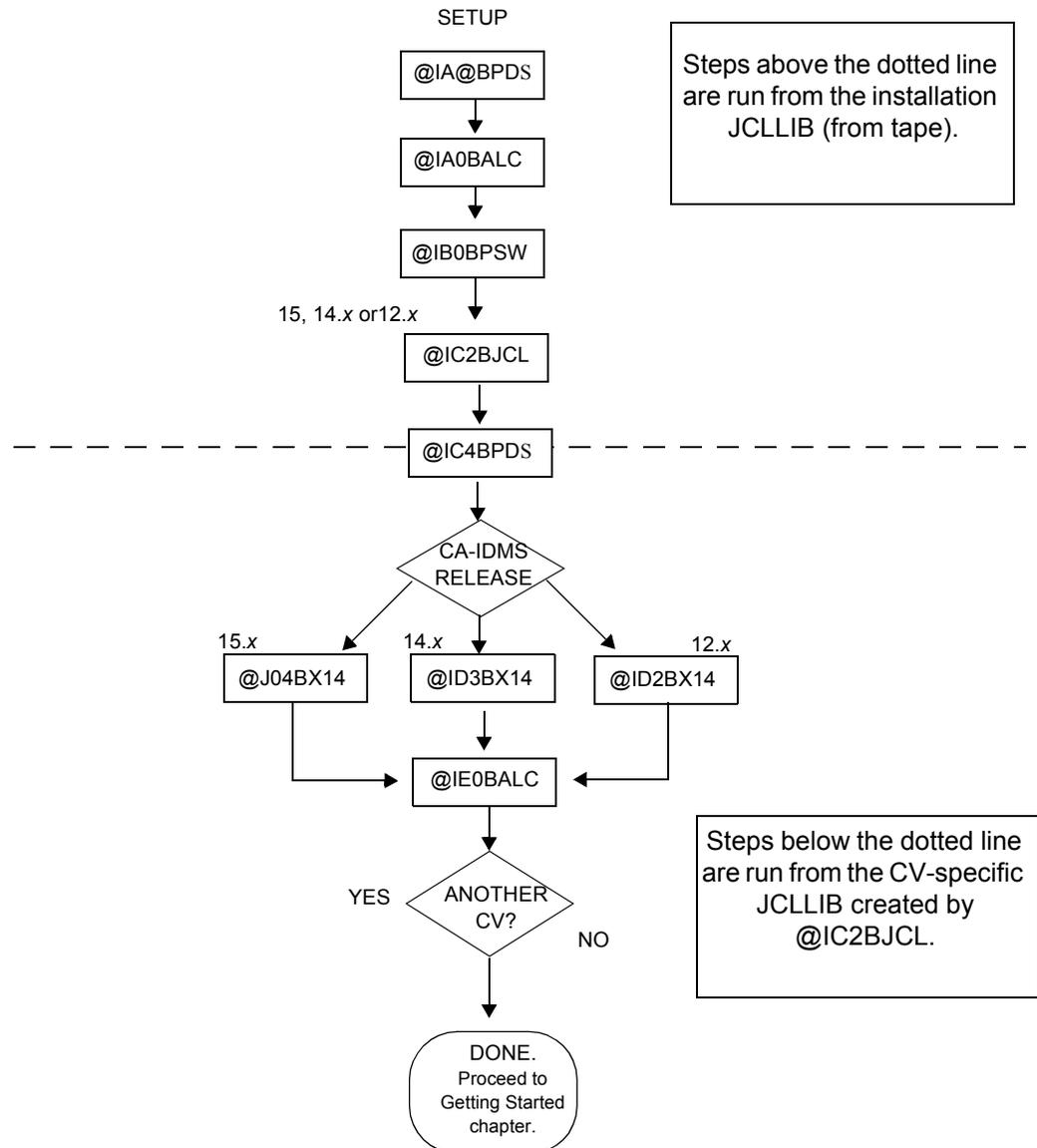
- The sixth, seventh, and eighth characters are three-digit JCL identifiers.

For example, the job name @IA0BALC would be interpreted as the first in a series of install jobs (indicated by the IA) for a IDMS database (herein called IDMS) that is both network and relational for Release 12.x, and/or 14.x and 15.x (indicated by the 0B). ALC is merely the JCL identifier.

## Setup Steps

After you have installed Tracer, complete the steps described in the remaining sections of this chapter. These steps run several one-time setup jobs that prepare your system for running Tracer. An overview of the setup process is illustrated in [Figure 3](#).

Figure 3 • Setup Process Overview



### **Step 1 - Run PDSALTER**

**To Change User Required Names to Site Standards.** Run job @IA@BPDS to make all user-defined changes for dataset names to the JCL for all Tracer installation jobs. Depending on your release of IDMS, use member, PDSALT12 (CA-IDMS Version 12.0), PDSALT14 (CA-IDMS Version 14.0), or PDSALT15 (CA-IDMS Version 15.0) to install Tracer.

See the JCL notes for explanation of the fields to change.

### **Step 2 - Run Library Allocation Job**

Run job @IA0BALC, found in the JCL library uploaded from the installation tape. This job allocates these site-specific libraries:

**PASSWORD.** This is the password library to store the load modules that provide authorization to use Tracer.

**MAINTNCE.** This is the maintenance library to store any load modules that have been changed by PTFs or enhancements that are received at a later date.

### **Step 3 - Run Product Authorization Job**

Run job @IB0BPSW, found in the JCL library uploaded from the installation tape. This job authorizes Tracer to run at your site. In this step, you must provide information found in the authorization letter provided with your installation materials.

Before running the @IB0BPSW job, make sure you have an appropriate job card for your site. If you have more than one ASG product that uses the ASGPTBL module, you can create one module that contains all ASG product authorization codes and maintain it in a separate load library that is accessible to all ASG products.

### **Step 4 - Allocate CV-specific Datasets**

**For CA-IDMS 12.x, 14.x, or 15.x Users Only.** If you are running CA-IDMS 12.x, 14.x, or 15.x, run job @IC2BJCL, found in the JCL library uploaded from the installation tape. This job accomplishes these tasks:

- Creates a CV-specific JCL library
- Copies the Tracer steps necessary to run against a CA-IDMS 12.x or 14.x, or 15.x environment
- Leaves the Tracer JCLLIB uploaded from the installation tape intact

In the JCL, TCR.Rxxx.JCLLIB is the JCL library uploaded from the installation tape. TCR.RXX.CV#xxx.JCLLIB is the site-specific JCL library created by this job.

### **Required Action: Library Change**

Run all jobs from this point forward from the CV specific JCLLIB created in this step.

**Caution!** *Do not* run the following Tracer jobs from the JCLLIB uploaded from the installation tape.

### **Step 5 - Run PDSALTER to Change CV Specific Dataset Names**

**For CA-IDMS 12.x, 14.x, or 15.x Users Only.** Run job @IC4BPDS if you need to change the dataset names in the CV specific JCLLIB to a CV number other than the one used in setup job @IA@BPDS.

This job is optional; use it only if you are running Tracer on more than one CV. If you are running Tracer on one CV, the CV identifier in the dataset names has already been changed in the setup job @IA@BPDS. This job can be used for subsequent runs of Tracer on other CVs.

The explanation for the field to change is in the JCL of the job.

### **Step 6 - Install Tracer Exit 14**

**For CA-IDMS 12.x, 14.x, or 15.x Users Only (Optional).** If you are running CA-IDMS release 12.x use job @ID2BX14, for release 14.x use job @ID3BX14, and for 15.x use job @ID4BX14. These jobs contain JCL to install Exit 14 into a CA-IDMS 12.x, 14.x, or 15.x CV.

The installation of Exit 14 is optional; it need not be installed to run Tracer. However, Exit 14 must be installed prior to running @TE0BREF if you wish to report on user ID, terminal name, task name, and subschema name. Exit 14 must also be installed if you wish to use the exit-related fields as selection criteria (see ["Selection Criteria Masking" on page 55](#)).

The CA-IDMS System Exit 14 is included as both a load module and an Assembler language source module as follows:

For release 12.x

Load module: TCR.Rxxx.LOADLIB(I12XUX14)  
Source module: TCR.Rxxx.JCLLIB(I12XUX14)

For release 14.0

Load module: TCR.Rxxx.LOADLIB(I14XUX14) or (I141UX14) for 14.1  
Source module: TCR.Rxxx.JCLLIB(I14XUX14) or (I141UX14) for 14.1

For release 15.0

Load module: TCR.Rxxx.LOADLIB(I150UX14)  
Source module: TCR.Rxxx.JCLLIB(I150UX14)

If your site does not use the standard IDMS sign-on procedure, you may have to make some modifications for this exit to work at your site. The exit is set up to work for each BIND RUN UNIT for DC, DDS, UCF, and batch sign-ons. The exit has documentation in the code pertinent to the offset that might affect your site.

If your site is not currently using the CA-IDMS System Exit 14, you may not need the source module. If your site has written code that is currently running as Exit 14, you may take the RHDCUX14 source provided and insert it into your own code. Thus, you maintain your existing functionality and supplement it with that provided by Tracer.

**Note:** \_\_\_\_\_

The ASG exit contains the IDMS MACRO #RTN, which returns directly to IDMS.  
\_\_\_\_\_

### ***SYSGEN Considerations***

- If the CWASIZE is set to exactly 500, then the Tracer Exit 14 will use the 250 bytes after the initial 250 bytes of the CWA (Common Work Area) for internal diagnostic purposes. You should only have your CWASIZE set to exactly 500 if technical support personnel requests this for diagnostic reasons.
- The CA-IDMS System Exit 14 uses approximately 170 bytes of stack area. Therefore, the stack size should be large enough for this extra usage.

The CV must be recycled before the CA-IDMS System Exit 14 becomes active.

**Note to Trial Customers:**

You may trial 99 percent of this product's features without installing this exit.

### ***Step 7 - Allocate CV-specific Datasets***

**All IDMS users.** Run job @IE0BALC, found in the CV-specific JCL. This job allocates these CV-specific libraries, which will be used by subsequent Tracer jobs:

**DBTRECIN.SORTED.** This library contains sorted, concatenated DBTRECIN DMCL, record, and set.

**DBTRECIN.DMCLST.** This library contains DMCL DBTRECIN records.

**DBTRECIN.RECDES.** This library contains record and set DBTRECIN records.

**SCHEMAPS.LOADLIB.** This library contains the schema record element map.

**IDMSRPTS.RECDES.** This library contains the output of IDMSRPTS record and set DESC.

**IDMSRPTS.DMCLST.** This library contains the output of the IDMSRPTS DMCL listing.

**RECELEM.MACLIB.** This library contains the record element macros.

**SQL.RPT2.** This library contains the output of SQL Report 002.

**CNTL.** This library contains control cards.

**DBTUNFIN.** This library contains unfinished run units.

**Note:** \_\_\_\_\_

If you are running Tracer on multiple CVs, you must complete steps 4 through 10 for each individual CV.

\_\_\_\_\_

When you have completed all installation and setup steps, proceed to ["Getting Started with Tracer" on page 19](#).

---

# 3

## Getting Started with Tracer

---

This chapter explains the run sequence of the Tracer product and contains these sections:

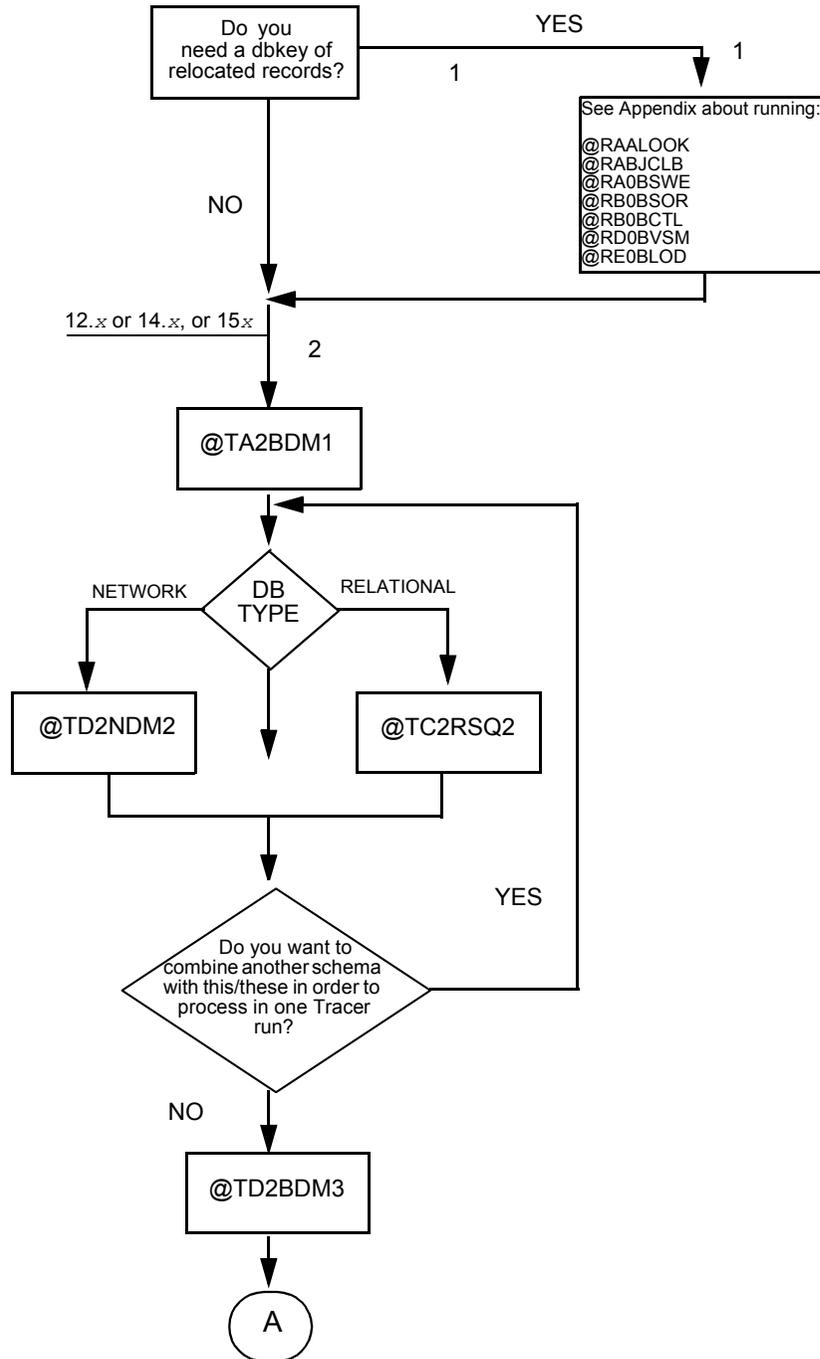
Section	Page
<a href="#">Step 1 - Create Dbkey of Relocated Records</a>	<a href="#">21</a>
<a href="#">Step 2 - DMCL and Schema Mapping</a>	<a href="#">21</a>
<a href="#">Step 3 - Journal Reformat and Reports Job</a>	<a href="#">25</a>
<a href="#">Step 4 - Remove Rolled-back Run Units Job</a>	<a href="#">27</a>
<a href="#">Step 5 - Read Reformatted File and Reports Job</a>	<a href="#">29</a>
<a href="#">Step 6 - Run Unit Statistics Job</a>	<a href="#">29</a>

---

The program's run sequence involves running several jobs from the installation JCLLIB and the CV-specific JCLLIB created during setup. The appropriate actions are discussed in the following steps.

Figure 4 provides a graphical illustration of steps 1 and 2 and Tracer's run sequence. The A at the bottom of the graphic shows where this flowchart connects to the one shown in Figure 5 on page 24.

Figure 4 • Tracer Run Sequence: Steps 1 and 2



## Step 1 - Create Dbkey of Relocated Records

(This is an optional step) If you need a database key (dbkey) of relocated records, refer to ["Creating a Dbkey of Relocated Records" on page 197](#) for instructions on running these jobs:

```
@RAALOOK  
@RABJCLB  
@RA0BSWE  
@RB0BSOR  
@RC0BCTL  
@RD0BVSM  
@RE0BLOD
```

**Note:** \_\_\_\_\_

Most sites that run Tracer for audit and performance tuning purposes will not need to run these jobs that involve dbkeys for relocated record occurrences.

---

## Step 2 - DMCL and Schema Mapping

### Run Job @TA2BDM1

First, run job @TA2BDM1. This job is found in the CV-specific JCLLIB that was previously created by setup job @IC2BJCL (under ["Step 4 - Allocate CV-specific Datasets" on page 15](#)).

@TA2BDM1 runs IDMSRPTS to obtain global DMCL information and create the DMCL DBTRECIN file. Run this job once for the global DMCL in a specific CV.

**Note:** \_\_\_\_\_

If you need a signon statement to specify a user ID and password to the IDMSRPTS utility, you should add the necessary line in the SYSIPT DD control statement in step RPTSDMCL.

Error Condition: System completion code S130 indicates an invalid dictionary name for the DMCL name specified.

---

## Run Job @TB2NDM2

**For CA-IDMS 12.x, 14.x, or 15.x Network Environments only.** If your CA-IDMS 12.x, 14.x, or 15.x environment is a network environment, run job @TB2NDM2, which is found in the CV-specific JCLLIB created during setup. @TB2NDM2 does the following:

- Runs IDMSRPTS to obtain record and set information
- Formats that information into a RECDDES DBTRECIN file
- Creates macros for the schema mapping load module.

You must run this job once per schema in a particular CV environment.

Run the @TB2NDM2 jobstream for each schema that defines records updated by the CV that produced the journal archive tapes against which you will later run Tracer.

**Note:** \_\_\_\_\_

Error Condition: IDMS return code 1111 indicates an invalid dictionary name for the schema name specified or the specified schema is not found.

---

After running this jobstream for each schema that contains records you want Tracer to process, run the jobstream @TD2BDM3 (see "[Run Job @TD2BDM3](#)") to concatenate all of the DBTRECIN files for the schemas you ran this job to produce.

You must run this job again for a particular schema whenever you do a restructure or an unload/reload (i.e., change the physical layout of the elements in a record or add a new record).

## Run Job @TC2RSQ2

**For CA-IDMS 12.x, 14.x, or 15.x Relational Environments only.**

If you are running Tracer in a CA-IDMS 12.x, 14.x, 15.x relational environment, run job @TC2RSQ2, which is found in the CV-specific JCL library created during setup. @TC2RSQ2 runs CULPRIT to obtain record and set information, formats that information into a RECDDES DBTRECIN file, and creates macros for the schema mapping load module. Run this job once for each schema you wish to process as part of a group of schemas in @TA1NDM2 (see the next section, "[Run Job @TD2BDM3](#)").

## Run Job @TD2BDM3

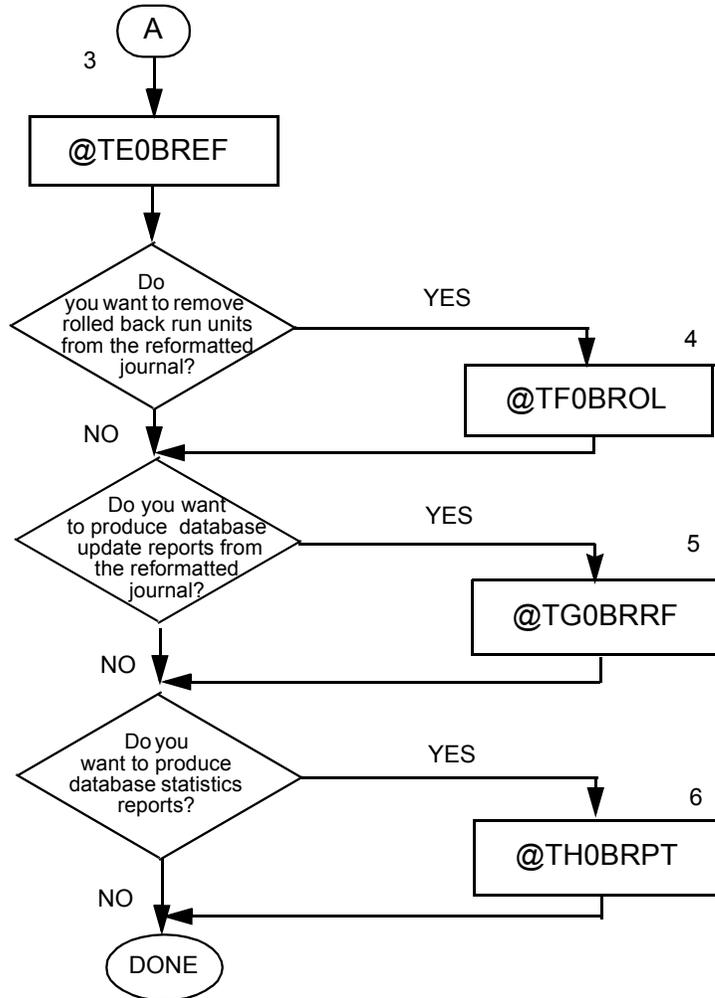
Regardless of whether your CA-IDMS 12.x, 14.x, or 15.x environment is network or relational, run job @TD2BDM3. This job is found in the CV-specific JCLLIB created during setup.

Job @TD2BDM3 sorts the global DMCL DBTRECIN file with one or more RECDES DBTRECIN files, then assembles and links the non-executable schema mapping load module.

@TD2BDM3 builds a DBTRECIN file containing information about records in more than one schema. This file will be used when Tracer reads the journal archive data set(s). Then it can process records from several different schemas in a single pass of the journals. This job also builds a MULTISCH schema load module containing information about each record from the schemas that were specified earlier (i.e., when you ran the @TB2NDM2 or @TC2RSQ2 jobs).

Figure 5 shows a graphical illustration of steps 3 through 6. The A at the top of the flowchart shows where the flowchart shown in Figure 4 on page 20 connects to this one.

Figure 5 • Tracer Run Sequence: Steps 3 through 6



## Step 3 - Journal Reformat and Reports Job

DBTRFMT is the journal reformat program. This program reads and reformats the journal archive file(s) and produces reports and/or a reformatted journal file. This is the core component of Tracer and its main purpose is to de-span, de-fragment, de-relocate, and/or decompress the journaled images stored in the offloaded IDMS journal file(s).

Input to this program includes the following:

- Journal archive file
- DBTINPUT control cards
- DBTRECIN schema map

The journal archive file that serves as input to this program is an offloaded file (the output of the IDMSAJNL utility), not an active journal file. The DBTINPUT control cards allow you to override any default values for selection criteria (e.g., which reports to generate, the number of journal records to process, which reformatted files to generate, etc.).

The DBTRFMT program can generate many output lines. Control parameters help pinpoint the date and time of updates to process, range of run units to process, and many other factors that control and limit both the records processed and the output of this program.

SQL parameters greatly enhance your ability to control processing and output by allowing you to make use of boolean logic and masking as part of your selections. The control parameters and SQL syntax is explained in ["Control Parameters" on page 37](#).

The main output from the DBTRFMT program includes the following:

- Tracer reports
- Reformatted journal file.

You may also produce files containing information on unfinished run units, DBOL/DMLO updates, SQL-selected journal images, aborts, stamp information, and run units.

Through the DBTINPUT control parameters, you can request one or more of these reports:

**Report 01.** Track Physical Record Updates (field-level)

**Report 02 DBOL/DMLO.** Updates to Any Subschema

**Report 05.** Track Set Pointers and Record Data Updates

**Report 11.** Track Physical Record Updates (record occurrence level)

**Tracker Field-level Audit Report.** This report displays the values of each record element field in the updated record occurrences.

For a detailed explanation of these reports, refer to ["Tracer Report Facility" on page 101](#).

## Run Job @TE0BREF

DBTRFMT is executed by running job @TE0BREF, which is found in the CV-specific JCLLIB created during setup. This job reads the journal archive files and produces the reformatted journal file and, optionally, the Tracer reports.

## Output Files

DBTRFMT outputs these files:

**DBTRFMTJ.** You may choose to have DBTRFMT generate a reformatted journal file. The result is a programmer-friendly version of the IDMS journal (a record layout is provided). This file may serve as input to user-written applications reporting programs. You can even allow Tracer to be the driver that controls the calls to your program.

This file may also be used as input to site-written programs that perform incremental updates of a non-IDMS parallel database. This is a more efficient approach than totally rebuilding the parallel database. These are only two of many possible uses of the reformatted journal file. As you become familiar with the information provided by the reformatted file, you may discover many creative solutions to your site's business, technical, and performance needs.

**DBTUNFIN.** The dataset you specify for the DBTUNFIN ddname contains the unfinished run units for any run. Use this file if you use Tracer to process non-quieted journals. For more information, see ["Specialized and Advanced Topics" on page 31](#).

**DBOLDMLO.** The dataset you specify for the DBOLDMLO ddname contains the reformatted journal images of records updated by either ASG-DBOL or DMLO only when you request that Tracer build a DBOLDMLO file. This provides a much-needed audit trail for ad hoc changes made through DBOL or DMLO.

**DBTRFMTS.** The dataset you specify for the DBTRFMTS ddname contains the reformatted journal images of records updated only if they matched the SQL specified selection criteria and hence would be included on the Tracker report.

**DBTABRTJ.** The dataset you specify for the DBTABRTJ ddname contains the end of run unit information to be subsequently processed by the DBTROLBK program to remove records for recovery units that either aborted or were still in progress when Tracer processed the last journal image on the journal archive file (JRNLARCH). This file is input to the JCL in member @TF0BROL.

**DBTSTAMP.** The dataset you specify for the DBTSTAMP ddname contains the user journal records written by the Tracer provided System Exit 14 (RHDCUX14).

**DBTRUNUI.** The dataset you specify for the DBTRUNUI ddname contains records that show all the information about a run unit at the time of an ENDJ or ABRT record, including a table showing each update run unit that was still active when this run unit completed.

You may dummy out these files if not needed for processing at your site: DBOLDBLO, DBTRFMTS, DBTSTAMP and DBTRUNUI.

## Step 4 - Remove Rolled-back Run Units Job

The DBTRFMT program reads and reformats the IDMS journal archive file(s). It performs this task for each journal record encountered in the file that meets the selection criteria. This presents an interesting problem, as IDMS automatically rolls back any changes made if the updating program abends. Not only does IDMS do this rollback automatically, but it gives programmers the ability to manually perform a rollback of changes through the ROLLBACK DML verb. This may result in a report identifying changes that were later rolled back. You may also have a reformatted file that also identifies changes that were later rolled back. DBTROLBK handles these conditions.

DBTROLBK effectively removes all record updates that IDMS rolled out because of an abort or an application program's use of the ROLLBACK DML verb. The first step in the job sorts the DBTABRTJ file produced earlier when the DBTRFMT program was run.

Input to the sort step includes:

- DBTABRTJ file
- Sort parameters
- DBTUNFIN

Output from the sort step is a sorted ABRTJ file. The abort data must always be sorted by the journal sequence number (not of the ABRT or ENDJ record, but of the BGIN record for that run unit) before running DBTROLBK.

### **Run Job @TF0BROL**

DBTROLBK is executed by running job @TF0BROL, which is found in the CV-specific JCLLIB created during setup. This job reads the reformatted journal file (DBTABRTJ) previously built by the DBTRFMT program (job @TE0BREF) and writes a new reformatted file (DBTRFMTO) that contains only the updates that IDMS did not roll back (because of an aborted recovery unit).

## DBTINPUT Control Cards

The program parameters will be entered in the DBTINPUT control cards. The DBTINPUT control cards allow you to override any default values for selection criteria (e.g., which reports to generate, the number of journal records to process, which reformatted files to generate, etc.).

## Input Files

Input files to DBTROLLBK are listed below.

**DBTABRTJ.** This file is created by running the program DBTRFMT. It contains job information about aborted run units (ABRT type journal records) and run units that ended successfully (ENDJ type journal records). DBTROLLBK assumes that no more than 100 run units will ever be in progress.

**DBTRFMTI.** This dname designates the file that contains reformatted journal records. This is the reformatted journal archive file produced in @TE0BREF. It may be either the file specified by //DBTRFMTJ, //DBTRFMTS, or //DBOLDLMLO.

**DBOLDMLO.** You may use the DBOLDMLO file if you wish to rollback updates performed by an aborted DBOL or DMLO online session.

## Output Files

These are the files output by DBTROLLBK:

**DBTINPRG.** This file holds all update run units in progress at the time the IDMS journal was offloaded. In other words, no ABRT or ENDJ records were found for this recovery unit in the present journal. For detailed information, see ["Specialized and Advanced Topics" on page 31](#).

**DBTRFMTO.** This file contains all reformatted journal records that were not rolled back in the database. If you are using Tracer to keep a parallel database in sync, you would use this file rather than the DBTRFMTJ file, to ensure that only permanent updates are applied to the parallel database.

**DBTROLLDB.** This optional output file contains all journal entries for updates that IDMS eventually rolled back.

## Step 5 - Read Reformatted File and Reports Job

The DBTRRFMT program reads the reformatted file that was purged of records associated with aborted recovery units by @TF0BROL. Since @TF0BROL was run first, reports produced by this job do not show any records that IDMS rolled back (due to the updating program aborting or specifying the ROLLBACK DML command).

**Note:** \_\_\_\_\_

The program executed here is DBTRRFMT (read reformatted) and the program executed in @TE0BREF is DBTRFMT (reformat).  
\_\_\_\_\_

### Run Job @TG0BRRF

To execute DBTRRFMT, run job @TG0BRRF, which is found in the CV-specific JCLLIB created during setup. This job reads the reformatted file now purged of records associated with the aborted recovery units by DBTROLBK (in @TF0BROL) and produces Tracer reports.

## Step 6 - Run Unit Statistics Job

**(Reports 30 - 41)**

DBTDDBART, Tracer's Run Unit Statistics program, reports on data that helps you tune your database. It also provides statistics on run units, such as concurrent run units, run unit resources, and association with record locks.

### Run Job @TH0BRPT

To execute DBTDDBART, run job @TH0BRPT, which is found in the CV-specific JCLLIB created during setup. This job reads the journal archive files and produces statistics and performance reports.

### Input File

JRNLARCH is the journal archive file that serves as input to DBTDDBART. JRNLARCH is an offload file output by the IDMSAJNL utility. JRNLARCH is not one of the active journal files.

### DBTINPUT Control Cards

The program parameters will be entered in the DBTINPUT control cards. The DBTINPUT control cards allow you to override any default values for selection criteria (e.g., which reports to generate, the number of journal records to process, which reformatted files to generate, etc.).

Response time is measured in hundreds of seconds for the DBTINPUT control parameters. This measurement of response time determines whether the run unit is included on the DBTRPT31 report.

## **Output Reports**

The DBTDBART program produces these output reports:

**Report 30 (DBTRPT30).** Concurrent Run Units Detail Report

**Report 31 (DBTRPT31).** Run Units Suspect Association with Record Locks

**Report 33 (DBTRPT33).** User Report by Subschema

**Report 34 (DBTRPT34).** Run Unit Report

**Report 37 (DBTRPT37).** Concurrent Run Unit Summary Report

**Report 38 (DBTRPT38).** Program Resource Report (all reports: 39, 40, and, 41)

**Report 39 (DBTRPT39).** Run Unit Resource Report by Program Name

**Report 40 (DBTRPT40).** Run Unit Resource Report by Most Used

**Report 41 (DBTRPT41).** Run Unit Resource Report by Descending Run Time

---

# 4

## Specialized and Advanced Topics

---

This chapter discusses specialized applications and topics intended for more experienced Tracer users, including information about:

Section	Page
<a href="#">Processing Non-quiesced Journal Archive Tapes</a>	<a href="#">31</a>
<a href="#">Splitting Reformatted Journal Records</a>	<a href="#">34</a>
<a href="#">Incremental Refreshing of Parallel Databases</a>	<a href="#">35</a>

---

### Processing Non-quiesced Journal Archive Tapes

*Non-quiesced* processing is undertaken when the beginning and/or ending of the journal file(s) that Tracer reads may contain run-units in progress in either of these states:

- The BGIN record has been written to a previous journal
- The ENDJ or ABRT record for the in progress run-unit is yet to be written to a subsequently-to-be-produced journal file

There are two files necessary for processing non-quiesced journals. The ddnames for these two files are: //MEMRYIN and //MEMRYOUT.

In order to understand the purpose for these files and what they accomplish, you should keep in mind a fundamental rule that governs how Tracer works. Tracer does not SORT all the detail records written to the journal. In large volume IDMS processing sites, this would place a big burden on the tape management systems and require additional processing time.

Instead, Tracer associates the run-unit information written to the journal with the individual updated records by using tables that are kept in memory and updated as new run-unit level information is encountered.

When a run-unit begins on the journal supplied to one run of Tracer, and the run-unit updates several records written to that journal, it creates a non-quiesced situation for Tracer to process. When Tracer reads a Journal, it must keep in memory all the information about each in progress run-unit that occurred from the end of one job to the beginning of another.

Tracer uses the //MEMRYOUT to write the contents of its memory tables when it completes processing a journal. So, the next time Tracer runs, it refreshes these tables by reading information from the //MEMRYIN file.

The file that is specified for the DDNAME //MEMRYOUT to TRACER-JOB-1 is the same file that is specified for the DDNAME //MEMRYIN to TRACER-JOB-2. ASG recommends using a single GDG for the dataset that is specified for both the //MEMRYIN and the //MEMRYOUT DDNAMEs, where //MEMRYIN is always one generation less than //MEMRYOUT.

You may specify two parameters in the //DBTINPUT file during the RUNRJRNL step that reads the journal. These are the parameters for non-quiesced journal processing:

```
SEL-YN-USE-MEMRYIN-FILE    Y    Y=REFRESH MEMORY TABLES FROM LAST
SEL-YN-USE-MEMRYOUT-FILE   Y    Y=WRITE MEMORY TABLES FOR NEXT RUN
```

The first Tracer job in a series of runs will have the parameter SEL-YN-USE-MEMRYIN-FILE set to N and the parameter SEL-YN-USE-MEMRYOUT-FILE set to Y. Each subsequent run will have both parameters set to Y, and each run will present to Tracer the next logical journal tape(s) produced by the particular IDMS-CV.

When these MEMORY parameters are used, Tracer writes additional information to the //MSGFILE to show the journal sequence numbers at the end and beginning of the journals that are input to Tracer. Also, when the SEL-YN-USE-MEMRYIN-FILE parameter is set to Y, and the first journal sequence number Tracer encounters is not a value of 1 greater than the last journal sequence number encountered on the previous run, Tracer outputs messages that indicate that the wrong journals seem to have been specified as input.

Tracer abends the job, allowing you to investigate why some journal activity has been skipped. An exception to this is when the journal sequence number is set to 1, so that it indicates to Tracer that an initialize journal operation was done by the site. In this case, Tracer outputs a message that shows that this has happened. The return-code is set to non-zero if the contents of the //MEMRYIN file indicate that there was at least one run-unit in-progress when the end of the Journal was encountered during the previous run. [Figure 6 on page 33](#) shows sample MEMRYOUT and MEMRYIN files.

**Figure 6 • Sample JCL f- MEMRYOUT and MEMRYIN files**

```
//MEMRYOUT DD DSN=TCR.R54.ASG.CV#121.MEMRY(+1),
//          DCB=(RECFM=VB,LRECL=22300,BLKSIZE=22304),
//          SPACE=(22304,(5,20),RLSE),
//          DISP=(,CATLG,DELETE),
//**       VOL=SER=DBS001,
//          UNIT=SYSDA
//MEMRYIN  DD DSN=TCR.R54.ASG.CV#121.MEMRY(0),DISP=SHR
```

You should set the DBTUNFIN parameters as shown in [Figure 7](#) in the step that reads the journals:

**Figure 7 • DBTUNFIN Parameter Settings**

```
//DBTINPUT DD *
...
YN-READ-DBTUNFIN          N WHEN USING MEMRYIN - THIS SET 'N'
YN-WRITE-DBTUNFIN         Y WRITE 1 REC TO //DBTUNFIN FILE
...
* -----
* -B- THE FOLLOWING PARAMETERS ARE FOR NON-QUIESCED JOURNAL PROCESSING:
*     WHERE RUN-UNITS ARE IN-PROGRESS ACROSS INPUT JOURNAL BOUNDARIES.
SEL-YN-USE-MEMRYIN-FILE   Y  Y=REFRESH MEMORY TABLES FROM LAST
SEL-YN-USE-MEMRYOUT-FILE  Y  Y=WRITE MEMORY TABLES FOR NEXT RUN
* -E- -----
/*
*-----*
```

In the step that runs program DBTROLBK (Job: @TF0BROL, Step: RUNROLBK), concatenate the DBTINPRG file that was produced the last time this step was run before the DBTRFMTJ file that was produced in the step that reads the journal.

**Figure 8 • DBTINPRG Concatenation (from prior run)**

```
//DBTRFMTI DD DSN=TCR.R70.CV###.DBTINPRG(0),DISP=SHR FROM LAST TIME
//          DD DSN=TCR.R70.CV###.DBTRFMTJ,DISP=SHR

//DBTINPUT DD *
...
YN-READ-DBTUNFIN                Y Y=USE //DBTUNFIN FROM @TE0BREF
...
/*
```

## Splitting Reformatted Journal Records

Occasionally, a site might need to segregate different reformatted journal records of the same system into two different reformatted journal files. Each group of end users would be able to run any Tracer report against its respective reformatted journal files.

The sample program name is DBTFLVL5. This shows how to split the journal record into two files based on values of a field. If you want to split based on subschema name, program name, user ID, and certain other key values, you might find it easier to run the DBTRFMT utility twice. You would specify one subschema name for the first run and another subschema name for the second run—creating two files.

The sample program DBTFLVL5 puts records and elements beginning with the prefix ASG into one reformatted file, and records and elements beginning with GAS into another reformatted file. Each output file could then be protected by your security system and naming conventions. You could then run Tracer reports against your own already-split files. Or, you could use sample program DBTCOBRW as a model and modify it to be a splitter. The program currently reads the reformatted journal file and writes it back to another file. However, the reformatted records could be written to two or three separate files.

## Incremental Refreshing of Parallel Databases

Tracer allows site-coded applications to use this technique to keep parallel copies of non-IDMS files synchronized with their IDMS based master file counterpart.

Tracer accomplishes the incremental refresh process by periodically applying certain changes to non-IDMS files. This includes only changes to specific record occurrences that have been updated on the master IDMS database since the last incremental refresh or the most recent complete rebuild of the non-IDMS file. The Tracer reformatted IDMS journal archive file contains both before and after snapshot images of only the IDMS record occurrences that were changed. User-defined applications may use the Tracer reformatted IDMS journal archive file as input to a program that actually does the incremental refresh maintenance operation to the non-IDMS file containing the parallel IDMS data.

Incremental refreshing is more advantageous than re-extraction and rebuilding the entire non-IDMS parallel file. Incremental refreshing requires significantly less physical I/O resources and hence requires less elapsed time to accomplish the *keep in sync* function. You can accomplish an incremental refresh immediately following each IDMS journal offload process. This ensures that data values in the non-IDMS parallel file are never over a couple of hours old.

Because of the extensive use of embedded physical pointers in the IDMS Network database, many sites choose to include the appropriate IDMS dbkey as a field in each row of the non-IDMS database tables and place an index on this field. The corresponding data in the relational non-IDMS file can then be updated when a CONNECT or DISCONNECT update verb is issued on the IDMS database. For example, only the pointer portion of IDMS records are updated during processing of certain CONNECT or DISCONNECT verbs.

You can configure Tracer to record all before and after pointer values in the reformatted file. Keeping the associated dbkey in the relational non-IDMS database is also important when MODIFY verbs are issued to a record whose logical key is not stored in that specific record occurrence. Instead, Tracer stores the logical key in a record occurrence of another record type that participates with it in a SET relationship as an owner record. An example of this would be when an EMPLOYEE record participates in the Employee-Salary set as an owner record and the MEMBER record type salary participates as a junction record in the Employee-Salary set and the Job-Salary set.

If a salary record occurrence is modified, it does not necessarily guarantee that the OWNER EMPLOYEE record occurrence will also be updated. Hence, the only way for the site-coded incremental refresh program to determine which row to update in the salary table, is to use the owner pointer dbkey from the Employee-Salary set that is associated with the updated IDMS SALARY record occurrence. This is why Tracer automatically returns the owner pointer for the VIA set in a special field. In the event that incremental refresh program needs pointers for other sets, as mentioned above, Tracer allows for all pointers to be placed in the reformatted file.

If you chooses to keep the physical dbkeys from the IDMS database in the non-IDMS parallel database, the parallel database must be completely rebuilt after an unload/reload of the IDMS master database.

### **Sample Programs**

See members EMPLRFMT and EMPLROLL in your SRCLIB. You could use these programs as models for setting up your own incremental refresh programs.

---

# 5

## Control Parameters

---

This chapter describes the control parameters required for the various Tracer functions and tasks. This chapter describes these types of parameters:

Section	Page
<a href="#">Run Control Parameters</a>	<a href="#">37</a>
<a href="#">Report Control Parameters</a>	<a href="#">39</a>
<a href="#">Default Input File (Tracer) Parameters</a>	<a href="#">72</a>
<a href="#">Field-level Selection Parameters</a>	<a href="#">73</a>
<a href="#">SQL Selection Parameters</a>	<a href="#">76</a>
<a href="#">Selection Criteria</a>	<a href="#">91</a>

---

### Run Control Parameters

#### ***Controlling How Tracer Calls Database Procedures to Decompress Journal Records***

If the SEL-YN-USE-MINI-CV-FOR-DBPS parameter is set to Y, Tracer will allow use of IDMS macros (e.g., #GETSTG) in the database procedures. If Tracer uses presspack or home-grown database procedures to decompress a record, it will start a local-mode (mini-cv) environment using the IDMSNWKA subschema. It is necessary to put the IDMS Load Library in the //STEPLIB file and specify a DMCL in the //SYSIDMS file.

```
SEL-YN-USE-MINI-CV-FOR-DBPS n
```

where  $n = Y$  or  $N$ .

## Defining the Parameters

This section lists and describes the run control parameters (and any alternate parameters that may be used).

### **NBR-ABRT-RECS-TO-DISPLAY**

Prompts DBTRFMT to issue COBOL DISPLAY statements showing the contents of all fields in the record for the specified number of run units when the run unit aborted.

### **NBR-CKPT-RECS-TO-DISPLAY**

Prompts DBTRFMT to issue COBOL DISPLAY statements showing each field name and its value for the specified first number of records written to the DBTRUNUI file.

### **NBR-PRSD-RECS-TO-DISPLAY**

Prompts DBTRFMT to issue COBOL DISPLAY statements showing each field name and value for the specified number of records written to the DBTRFMTJ (reformatted) file. Display statements are also printed for each field returned to the calling COBOL program from the DBTRACE Assembler subprogram.

### **NBR-STMP-RECS-TO-DISPLAY**

Prompts DBTRFMT to issue COBOL DISPLAY statements showing each field name and its value for the specified first number of records written to the DBTSTAMP file. This file contains the information that the Tracer Bind/Ready System Exit 14 writes to the IDMS journal.

### **NBR-SUSP-RECS-TO-DISPLAY**

Prompts DBTRFMT to issue COBOL DISPLAY statements that show each field containing information about run units that waited for another run unit due to contention over at least one locked dbkey.

### **NBR-TO-PROCESS**

Specifies the total number of journal records to be read before ending the job.

### **NBR-TO-REPORT**

Limits the number of records represented on any of the detail reports requested (e.g., 01, 05, or 11). The job does not stop after this number of records has been placed on the requested reports. The reports stop producing additional lines of output and the reformatted file continues to be built until either the end of the journal archive file is encountered or the NBR-TO-PROCESS number of records have been read.

### **TRKR-YN-USE-FLDSELS**

Determines which records are printed on the Tracker report when Tracer reads the //FLDSELS input file to determine which SQL-like statements are to be used. Synonyms are YN-FLDSELS and YN-USE-FLDSELS.

### **YN-DISPLAY-FLDSELS-TRACE**

Debugs/traces SQL selection subprogram.

**YN-DISPLAY-SELECTOR-TRACE**

Debugs/traces field selection subprogram.

**YN-DISPLAY-TRACE**

Debugs/traces main program.

**YN-MIXED-CASE**

Indicates if the headings on the reports should be printed in both upper and lower case. If this is set to N, then all headings are in upper case letters.

**YN-READ-DBTUNFIN**

Instructs Tracer to use the DBTUNFIN file as input, thereby processing all unfinished run units from the prior journal.

**YN-USE-DEFAULT-FILE**

Instructs Tracer to read and use the parameters supplied by the DEFAULTS file.

**YN-WRITE-DBTUNFIN**

Instructs Tracer to use the DBTUNFIN file as output. Stores all journal records from unfinished run units. At the time of processing, it is unknown whether the run unit will end or abort.

## Report Control Parameters

Report Control parameters fall into these seven categories:

- Parameters Affecting Printing Data in Hex
- Parameters Affecting Pointer-only Processing
- Parameters Affecting the Tracker Report
- Parameters Affecting Site-written Reports
- Parameters Affecting Restructured Records
- General Report Control Parameters
- Parameters for Selection Criteria.

## Parameters Affecting Printing Data in Hex

This section lists and describes all of the parameters that affect printing data in hex format. (These parameters apply to Report 11 only.)

### RECRPT-COLUMNS

Determines whether to use the horizontal dump format. For example, if you limit the report to 132 columns, you may display up to a 36-byte field (horizontal format takes 36 bytes for the character display, 72 characters for the hex display, and 19 characters for the line-type and general information, equaling 127 characters).

Example Horizontal Format:

```
DA BFOR DATA DISP: JOHN DOE CORP  HEX1:   D1D6C8D540C4D6C540C3 D6D9D740
```

Example Vertical Format:

```
DA BFOR  100 DATA: JOHN DOE CORPORATION
DA          HEX1: DDCD4CDC4CDDDDDDCECDD4
DA          HEX2: 168504650369769139650
          ....+.....1.....+.....2..
```

In the above examples, DA (data) is the line type. BFOR indicates a before data image. AFTR indicates an after data image.

In the vertical format, the number printed before DATA indicates how many hundreds of bytes this line is into the record. For example, a 2000-byte record requires several pages to print. Use this number with the optional scale line to identify the byte positions within this line.

Synonym: PHYSREC-RPT-COLUMNS

### RECRPT-DBKEY-DISPLAY-MODE

Indicates which dbkeys are printed (if any). These are the valid values:

- 0 = No dbkeys printed
- 1 = All dbkeys printed
- 2 = Only changed dbkeys printed

Synonym: DBKEY-DISPLAY-MODE

**RECRPT-HEX-DISP-FORMAT**

Indicates the dump format. These are the valid values:

- V = Vertical
- H = Horizontal
- B = Best fit (The program determines whether to print vertically or horizontally, depending on the number of bytes being dumped.)

Sometimes the program overrides the horizontal (H) option. For example, if you choose RECRPT-HEX-DISP-MODE option 2 (entire record), the program automatically prints the record in vertical format. The horizontal format saves paper when the item to be dumped is under 40 bytes.

**RECRPT-HEX-DISP-MODE**

Indicates which items are to be shown in display/hex format These are the valid values:

- 0 = Nothing displayed in display/hex dump
- 1 = CONTROL-KEY only (see definition below)
- 2 = Entire record
- 3 = Only dump from RECRPT-HEX-START-POS to RECRPT-HEX-STOP-POS
- 4 = Combination of 1 and 3 above

CONTROL-KEY is the minimum left-most part of the record that contains CALCKEYS or SORT-KEYS. For example, study this record:

```
01 DEMO-EMP-RECORD
02 EMP-SSN PIC X(9)
02 EMP-DEPT PIC X(4)
02 EMP-LASTNAME PIC X(25)
02 EMP-FIRSTNAME PIC X(20)
02 EMP-ADDRESS PIC X(36)
```

If EMP-SSN is the CALC-KEY and there is an integrated index on EMP-LASTNAME, the CONTROL-KEY would be the first 38 bytes of the record. The EMP-DEPT field would be included in the CONTROL-KEY though it is not actually a key.

**RECRPT-HEX-LAST-N-BYTES**

Prints the last specified number of bytes of each record (when a number greater than zero is specified) in the dump format specified by the previous parameters. This feature is primarily for sites that store a date/time/user ID stamp in the last few bytes of each record.

### **RECRPT-HEX-SCALE-LINE**

Indicates whether to print a scale line below the dump. These are the valid values:

- 0 = No scale line
- 1 = Print scale line after every dump (This makes a good separator between groupings of pointer and data records.)
- 2 = Print scale line only at the bottom of each page

### **RECRPT-HEX-START-POS**

Indicates which bytes of the record are to be printed when RECRPT-HEX-DISP-MODE is set to 3 or 4.

### **RECRPT-HEX-STOP-POS**

#### **RECRPT-HEX-LENGTH**

Limits the printing of a database record. This option is most meaningful when the report is limited to one record type.

For example, you could look at a IDMSRPTS listing and decide to print the salary record from byte 102 to byte 108. Rather than using RECRPT-HEX-STOP-POS, you may use RECRPT-HEX-LENGTH, which allows Tracer to compute the RECRPT-HEX-STOP-POS. You should use either the RECRPT-HEX-STOP-POS or the RECRPT-HEX-LENGTH parameter, not both.

### **RECRPT-YN-SHOW-CHANGED-BYTES**

Prints an extra line after each AFTR image, showing all bytes in the above display/hex dump that have changed (an asterisk prints below any byte that changed). This line is prefixed with the phrase KEY-CHANGE or MODIFIED-BYTES, depending on whether the dump is for the control-key versus a data range or for the entire record. Valid values are Y (Yes) or N (No).

## ***Parameters Affecting Pointer-only Processing***

A pointer-only journal record is a journal update in which pointers changed, but data did not. This could occur for these reasons:

- A CONNECT or DISCONNECT verb
- For adjacent records in sets when a STORE or ERASE of a member record in a set takes place
- For adjacent records in the set when a MODIFY DML verb was issued and the sort key was changed.

For example, if an EMPLOYEE record is stored in the DEPT-EMPLOYEE set, the next and prior pointers of the neighboring EMPLOYEE records must be updated (or if the EMPLOYEE record was first or last in the set, pointers of DEPT must be updated). These records are called non-target records because they were not the target of the DML verb. After these pointers are updated, the actual or target record is stored.

A target record is:

- A record that is stored, erased, or modified, and whose data changed.
- A record that is connected or disconnected, and whose next, prior, or owner pointers in a set were changed from null to actual pointers.

The DBTRFMT and DBTRRFMT programs default to avoid sending pointer-only records to the field-level selection subprogram for non-target updated records.

There are several additional parameters that allow you to determine whether pointer-only records are included on certain reports (i.e., passed to certain programs). The default for each of these parameters is N (No). If you set the variable to Y (Yes), you can base selection on impacted dbkeys. For example, you might be investigating a broken chain and the date/time it broke. You could list all updates to a specific dbkey number. These are the parameters:

#### **YN-PNTRONLY-RECS-TO-RECRPT**

Specifying Y for this parameter passes pointer-only records to all reporting programs except for the Tracker report program.

Synonyms: YN-PNTRONLY-RECS-TO-PHYSREC  
YN-PNTRONLY-RECS-TO-REPORT01  
YN-PNTRONLY-RECS-TO-REPORT05  
YN-PNTRONLY-RECS-TO-REPORT11

#### **YN-PNTRONLY-RECS-TO-SELECTOR**

Specifying Y for this parameter passes pointer-only records to the SELECTOR program.

Synonym: SELECTOR-RCVS-PNTRONLY-RECS

### **YN-PNTRONLY-RECS-TO-TRACKER**

Specifying Y for this parameter results in pointer-only records being passed to the Tracker report program.

Synonyms: YN-PNTRONLY-RECS-TO-TRACPRT

YN-PNTRONLY-RECS-TO-TRKR

### **YN-PNTRONLY-RECS-TO-USERRPT**

Indicating Y for this parameter results in pointer-only records being passed to a site-written report program.

## **Parameters Affecting the Tracker Report**

This section discusses the parameters and selection elements required for the field-level (Tracker) report.

### **TRACKER-SCHEMA-NAME**

Description of records that display on the Tracker report. The schema name must be supplied to produce a Tracker report. This is the name of the load module produced when running the jobstreams described in ["Step 2 - DMCL and Schema Mapping" on page 21](#).

Synonym: TRKR-SCHEMA-NAME

### **TRKR-DBKEY-DISPLAY-MODE**

Controls which pointer dbkeys are printed on the Tracker report (if any). These are the available settings:

- 0 = Do not print any dbkeys
- 1 = Always print all dbkeys
- 2 = Show only changed dbkeys

### **TRKR-RPT-YN-80-COL-MODE**

Stacks before and after images to make the report easy to read on an 80-column terminal.

Synonym: TRKR-RPT-YN-STACK

### **TRKR-RPT-YN-HDR-INC-ALWAYS**

Indicates whether the Tracker report should print the block of information typically printed for each updated record occurrence processed, even when no elements in that record qualify for printing.

For instance, if the TRKR-RPT-YN-INCLUDE-SAME parameter was set to N, the FLDINCL record ID was used to show that certain elements should not be printed for this record type, and a MODIFY verb changed those same elements, then no elements would be printed.

If TRKR-RPT-YN-HDR-INC-ALWAYS is set to N in this case, the header portion showing general information about the record occurrence does not print either.

**TRKR-RPT-YN-INCLUDE-GROUPS**

Includes or excludes group-level fields in the Tracker report. Valid values are Y (Yes) or N (No).

**TRKR-RPT-YN-INCLUDE-NULLS**

Includes or excludes fields that have a value of spaces or zero in the Tracker report. Valid values are Y (Yes) or N (No).

**TRKR-RPT-YN-INCLUDE-SAME**

For updates done by a MODIFY verb, includes or excludes fields where the before/after value did not change. Valid values are Y (Yes) or N (No). Setting this parameter to N reduces the number of pages required for the report.

**TRKR-USERPROG-NAME**

Specifies the name of a user-written program to be called when Tracer normally produces the Tracker report. The name of this program is the name of the load module of your exit program. Remember, your program must be in the STEPLIB concatenation to be loaded and executed at run-time. Tracer uses the COBOL DYNAM option when compiling Tracer programs.

If you would prefer the Tracker report to be formatted differently, you can produce your own Tracker-like report via this program. Or, you may write a reformatted journal archive file that has a different layout than the one automatically produced by Tracer (see members RFMTREC and RFMTRECL). This would be practical if your record layout contained only certain fields in the record and not the entire record occurrence.

**TRKR-YN-CALL-USERPROG**

Instructs Tracer to call a user-written program. You specify the program name via the TRKR-USERPROG-NAME parameter.

The installation tape provides a sample program, USERTRKR. This program writes to the ddname DBTRFMTS only those reformatted journal records that qualify for selection based on the Tracer SQL selection syntax.

**TRKR-YN-PRINT-REPORT**

Instructs Tracer to print the Tracker report. TRACKER-SCHEMA-NAME must be specified.

#### **TRKR-YN-USE-FLDINCL**

Indicates whether Tracer reads the input file //FLDINCL to determine which records and/or elements to include in the Tracker report.

Synonyms: YN-LIMIT-TRKR-EL-NAMES  
YN-LIMIT-TRKR-ELS

#### **WHICH-UTIL-PRINTS-TRKR**

(DBTRFMT or DBTRRFMT) Indicates which utility should print the Tracker report when running both utilities in the same job stream.

Synonym: WHICH-UTIL-PRINTS-TRACPRT

#### **YN-APPLY-SELECTOR-TO-TRKR**

Allows you to apply field-level selection to the Tracker report.

Synonyms: YN-APPLY-SELECTOR-TO-TRACKER  
YN-APPLY-SELECTOR-TO-TRACPRT

#### **YN-DISPLAY-TRKR-TRACE**

Debugs and traces the Tracker report program.

### ***Parameters Affecting Site-written Reports***

Tracer allows you the option of writing your own COBOL report programs if you have specific reporting needs that are not addressed by the product reports. You can even use Tracer as the driver that calls your report program. These parameters control and enhance the use of site-written reports:

#### **USERRPT-SUBPROGRAM-NAME**

Specifies the COBOL load module called to print a user-written report.

#### **YN-APPLY-SELECTOR-TO-USERRPT**

Allows you to apply field-level selection criteria to a user-written report.

#### **YN-DISPLAY-USERRPT-TRACE**

Debugs and traces user-written report subprogram.

### ***General Report Control Parameters***

This section provides information on these parameter types:

- Parameters affecting reports 01, 05, and 11
- Banner/Documentation feature
- //DBTINPUT parameters.

**LINES-PER-PAGE**

Indicates the number of lines printed per page. This allows you to take advantage of Form Control Buffers (FCB). These allow six or eight or more lines per inch.

**PRINT REPORT 01**

**PRINT REPORT 02**

**PRINT REPORT 05**

**PRINT REPORT 11**

Specify Y for any of these parameters to print the selected report.

**RECRPT-YN-DISPLAY-ALL-BATCH**

Excludes batch jobs from processing.

Synonym: RPT01-YN-DISPLAY-ALL-BATCH

**RECRPT-YN-STACK**

Adds a new line to the report after each existing line. This stack new line contains a 60-byte listing of each before and after image. This enables you to see more of the data portion of the changed record, while minimizing the amount of paper printed. (This is compared to printing the Tracker report that lists all elements changed on each record.)

Synonym: PHYSREC-RPT-YN-STACK

**SELECTOR-SUBPROGRAM-NAME**

Calls this COBOL load module to perform field-level selection criteria.

**PARAM-A1 through PARAM-A10**

Passes these parameters to user-written field selection programs.

**PARAM-N1 through PARAM-N10**

Passes these numeric parameters to user-written field selection programs.

**YN-APPLY-SELECTOR-TO-RECRPT**

Applies field-level selection criteria to reports 01, 05, and 11.

Synonyms: YN-APPLY-SELECTOR-TO-PHYSREC

YN-APPLY-SELECTOR-TO-REPORT01

YN-APPLY-SELECTOR-TO-REPORT05

YN-APPLY-SELECTOR-TO-REPORT11

**YN-DBKEY-LIMIT-TO-TARGET**

Instructs Tracer whether to print dbkeys if the record was not the target of the updating DML verb.

## **YN-DISPLAY-RECRPT-TRACE**

Debugs and traces the record report programs.

## **Documentation Banner Feature**

Reports 01 and 11 can optionally print a banner of report documentation at the top of the report, before the actual data listing. This banner makes reports easier to read. The banner is a text file stored in a PDS member (which you can update). The PDS member is printed at the beginning of the report. It contains summary documentation, including an explanation of each column on the report.

An optional four-line key to abbreviations footer can also be printed at the bottom of each page of the report.

If you want to use these header or footer features, you must include this JCL statement in your run-time JCL.

```
//BANRPT01DD DSN=XXXXX.YYYYY.SRCLIB(BANRPT01),DISP=SHR
//BANRPT11DD DSN=XXXXX.YYYYY.SRCLIB(BANRPT11),DISP=SHR
```

where *XXXXX.YYYYY* are the leading qualifiers of your Tracer-installed source dataset.

These two features are particularly useful for first time and occasional users of the product.

## **YN-PRINT-BANNER**

Y includes the banner/documentation page on reports.

## **YN-PRINT-FOOTER**

Y includes the key to abbreviations footer on reports.

## **Selection Criteria Parameters**

Selection criteria are the means by which Tracer allows you to specify attributes of the archived data so that the output contains only data relevant to your site.

Selection criteria is the initial step to Tracer operation. Using selection criteria, you can limit and tailor the types of records that are reformatted from the IDMS journal archives. a IDMS journal archive file containing information for only a single day often has over ten million individual record occurrences. Because of this volume of information, it is necessary to be able to sift through the material and to process only relevant updates. Selection criteria values limit the number of journaled records Tracer processes.

A COBOL copybook, called *DEFAULTS*, comes with Tracer (in your *SRCLIB*) to set the selection criteria default values. These values can be changed by recompiling the *DBTINPUT* program.

The recommended, more dynamic way to set defaults is to use the //DEFAULTS ddname. This ddname requires the same input parameters as //DBTINPUT.

Tracer selection criteria is classified according to the category of journaled updates excluded. There are two classifications of selection criteria used by Tracer to determine whether to process a specific update record occurrence: Run Unit Selection Criteria and Record Selection Criteria.

### **Run Unit Selection Criteria**

When Tracer encounters a BGIN record for a run unit, selection criteria tests are applied to determine when the subsequent BFOR or AFTR journal records for this run unit should be considered for processing. The parameters described below are run unit oriented:

#### **SEL-EXEMPT-SUBSCHEMA-MASK**

Identifies a subschema that should not be processed when you are processing more than one database segment on the same pass of the IDMS journal archive file. IDMS System Exit 14 is required for this criterion.

Default = NOEXEMPT.

#### **SEL-PROGRAM-MASK**

This parameter requires that the application program name being selected exists in the PROGRAM-NAME field of the COPY IDMS-SUBSCHEMA-BINDS copy book prior to the BIND run unit.

This parameter processes only updates requested by programs identified in the subschema control block.

Default = ALLVALUE.

#### **SEL-RU-BEGIN-DATE**

Specifying dates establishes a range for minimizing the data that must be traced to obtain the desired output. Through this parameter, all run units (i.e., anything that was updated on IDMS) that occurred on the given date are eligible for selection. Tracer compares this date parameter to the exact date/time stamp of the applicable run unit.

Enter the date (*MM/DD/YY*) of the first run unit to be traced or specify ALLDATES to process all run units, regardless of the date they began. If the begin and end date is the same, you must specify it for both SEL-RU-BEGIN-DATE and SEL-RU-END-DATE.

Default = ALLVALUE.

### **SEL-RU-END-DATE**

When you specify a begin date (SEL-RU-BEGIN-DATE), you must also specify the end date of the last run unit eligible for selection (i.e., the latest date associated with the begin time on the date/time stamp for the run units you want chosen). Tracer ignores this parameter if ALLDATES was specified for SEL-RU-BEGIN-DATE. If the begin and end date is the same, you must specify it for both SEL-RU-BEGIN-DATE and SEL-RU-END-DATE.

Default = *MM/DD/YY*.

### **SEL-RU-BEGIN-TIME**

Specifying time of day allows you to reduce the size of the data to be traced.

Tracer compares this parameter to the run unit date/time stamp. Enter 0 (i.e., all times) if you want to process run units regardless of what time they began. If the begin and end times are the same, you must specify that time for both the SEL-RU-BEGIN-TIME and SEL-RU-END-TIME parameters.

Specifying a time range limits the time of day that Tracer selects for each day in the date range. For example, if:

```
SEL-RU-BEGIN-DATE = 08/08/01
SEL-RU-END-DATE   = 08/24/01
SEL-RU-BEGIN-TIME = 0009000000
SEL-RU-END-TIME   = 0010000000
```

Tracer selects only times between 9 and 10 a.m. for each day from 08/08 through 08/24.

Enter the start time of the first run unit to be eligible for selection in this format:

*HHMMSSHS*

where:

*HH* = hours  
*MM* = minutes  
*SS* = seconds  
*HS* = hundredths-of-seconds

**Note:** \_\_\_\_\_

This is the begin time of the run unit, not the time of the record update. Use SEL-UPDATE-TIME-START to specify a time range for individual updates within a run unit.

\_\_\_\_\_

Default = 0.

**SEL-YN-INCLUDE-CENTURY-N-0****DATE**

If the SEL-YN-INCLUDE-CENTURY-N-DATE parameter is set to Y, Tracer will include the century portion of the update date (IDMS releases 12.X, 14.X, and 15.X) in the journal records.

This is an example of the parameter:

```
SEL-YN-INCLUDE-CENTURY-N-DATE n
```

where  $n = Y$  or  $N$ .

The RFJ-RUN-UNIT-START-DATE parameter in the reformatted file (//DBTRFMTJ) is populated with the format CCYYMMDD (instead of the default layout of MM/DD/YY). When the SEL-YN-INCLUDE-CENTURY-N-DATE parameter is set to Y, this redefinition is applicable for the RFJ-RUN-UNIT-START-DATE area of the reformatted record layout, as seen in [Figure 9](#).

**Figure 9 • SEL-YN-INCLUDE-CENTURY-N-DATE Parameter**

```
10 RFJ-START-DATE-WITH-CENTURY REDEFINES RFJ-RUN-UNIT-START-DATE.
15 RFJ-START-DATE-WC-CENTURY PIC 99.
15 RFJ-START-DATE-WC-YEAR PIC 99.
15 RFJ-START-DATE-WC-MONTH PIC 99.
15 RFJ-START-DATE-WC-DAY PIC 99.
```

Default = N.

**SEL-RU-END-TIME**

If you entered a specific begin time (SEL-RU-BEGIN-TIME), enter the ending time of the last run unit to be eligible for selection (i.e., the largest time associated with the beginning time of the date/time stamp). This parameter is ignored if 0 was entered for SEL-RU-BEGIN-TIME. If the begin and end times are the same, you must specify that time for both the SEL-RU-BEGIN-TIME and SEL-RU-END-TIME parameters.

Default = 0.

### **SEL-RU-ENVIRONMENT**

This defines the run unit environment to be processed. These are the five valid specifications:

- DBDC = Online run units only
- BATC = External run units, usually batch jobs
- CICS = Run units initiated by programs loaded in a CICS region
- ALL = All of the above
- [other] = If your systems programmer has used a system exit to put specific parameters in the ERE control block, specify that value here (e.g., CIC1 or CIC2).

Default = ALL.

### **SEL-RUN-UNIT-NUMBER-1**

Use this parameter if you selected O, R, or B in the TYPE-RUN-UNIT-SELECTION parameter. If you chose:

- O = Enter the required run unit ID here.
- R = Enter the lowest run unit in the range here.
- B = Enter one of the two run units here.
- A = This parameter is not used.

If you are selecting only one run unit, the values of the SEL-RUN-UNIT-NUMBER-1 and SEL-RUN-UNIT-NUMBER-2 must be the same.

Default = O.

### **SEL-RUN-UNIT-NUMBER-2**

Indicates the highest run unit ID in a range if you selected R for the SEL-TYPE-RUN-UNIT-SELECTION parameter. If you chose:

- O = This parameter is not used.
- R = This is the highest run unit number in the range.
- B = This is the second run unit number.
- A = This parameter is ignored.

If you are selecting only one run unit, the values of the SEL-RUN-UNIT-NUMBER-1 and SEL-RUN-UNIT-NUMBER-2 must be the same. The numbers cannot exceed a value of 4294967290.

Default = O.

### **SEL-USE ENDING TIME RECOVERY UNIT**

If the USE ENDING-TIME-RECOVERY-UNIT parameter is set to Y, Tracer will place the ending date and time of the recovery unit on the reformatted records. The USE ENDING-TIME-RECOVERY-UNIT parameter is specified in jobstream @TF0BROL, the job that removes the updates that IDMS automatically rolled-back off the database due to a transaction abort.

An example of the parameter is seen here:

```
USE ENDING-TIME-RECOVERY-UNIT n
```

where  $n = Y$  or  $N$ .

When the USE ENDING-TIME-RECOVERY-UNIT parameter is set to Y, this redefinition is necessary for the RFJ-START-DATE-TIME-FIELDS area of the reformatted record layout, as seen in [Figure 10](#).

**Figure 10 • USE ENDING-TIME-RECOVERY-UNIT Parameter**

```
05 RFJ-END-RECOVERY-DATE-TIME REDEFINES RFJ-START-DATE-TIME-FIELDS .
10 RFJ-DATE-CENT-YEAR-MONTH-DAY PIC 9(8) COMP-3.
10 RFJ-TIME-HR-MIN-SEC-MILLIONS PIC 9(12) COMP-3.
10 RFJ-DAY-OF-WEEK PIC X(2) .
10 RFJ-JULIAN-DAYS-INTO-YEAR PIC 9(3) COMP-3.
```

Default = N.

### **SEL-SUBSCHEMA-MASK**

Allows the user to specify a mask to limit the record occurrences Tracer processes to the run units that used the subschema specified. CA-IDMS System Exit 14 is required for this parameter to be used.

Default = ALLVALUE.

### **SEL-TASK-ID-NUMBER-1**

Use this parameter if you selected O or R in the SEL-TYPE-TASK-ID-SELECTION parameter. If you chose:

- O = Enter the required Task ID here.
- R = Enter the lowest Task ID in the range here.
- A = This parameter is ignored.

This number cannot exceed a value of 4294967290.

Default = O.

**SEL-TASK-ID-NUMBER-2**

Indicates the highest Task ID in a range when you have selected R in the SEL-TYPE-TASK-ID-SELECTION parameter.

If you chose:

- O = Enter the required Task ID here.
- A = This parameter is ignored.

If you are selecting only one Task ID, the values of the SEL-TASK-ID-NUMBER-1 and SEL-TASK-ID-NUMBER-2 must be the same. This number cannot exceed a value of 4294967290.

Default = O.

**SEL-TASK-NAME-MASK**

Limits Tracer processing to certain task codes. CA-IDMS System Exit 14 is required for this criterion. This criterion is not available for batch run units. Use ALLTASKS to bypass this selection criterion.

Default = ALLVALUE.

**SEL-TERMINAL-MASK**

Limits Tracer processing to certain physical Terminal IDs. CA-IDMS System Exit 14 is required for this criterion. This selection criterion is not available for batch run units. Use ALLTERMS to bypass this selection criterion.

Default = ALLVALUE.

**SEL-TYPE-RUN-UNIT-SELECTION**

Allows selection of a particular run unit or a range of run units to be processed. These are the available selections:

- A = All
- O = One
- R = Range
- B = Both

When this value is A, the program uses SEL-RUN-UNIT-NUMBER-1 through SEL-RUN-UNIT-NUMBER-2 (inclusive).

When this value is B, the program uses only SEL-RUN-UNIT-NUMBER-1 and SEL-RUN-UNIT-NUMBER-2.

Default = A.

#### **SEL-TYPE-TASK-ID-SELECTION**

Allows selection of a particular task ID or a range of task IDs to be processed. These are the available selections:

A = All  
O = One  
R = Range

When this value is not A, the program uses the SEL-TASK-ID-NUMBER-1 and SEL-TASK-ID-NUMBER-2.

Default = A.

#### **SEL-USERID-MASK**

Limits Tracer processing to particular user IDs. This parameter requires 32 characters that may be specified by either one blank space or an asterisk after the user ID. For example: ASG1 or ASG1\*. This criterion requires CA-IDMS System Exit 14.

**Note:** \_\_\_\_\_

For batch jobs, the user ID is often the jobname.  
\_\_\_\_\_

Default = ALLVALUE.

#### **SEL-YN-SKIP-CONCURRENT-CHKS**

Setting this parameter to Y reduces some of the CPU cycles required to run Tracer. If you do not want Tracer to try to determine which run units are waiting on other concurrently running run units in order to reduce dbkey locks, set this parameter to Y.

### ***Selection Criteria Masking***

A masking procedure exists for certain selection criteria where:

- An asterisk (\*) is a global mask character
- A question mark (?) is a positional mask character

Use Masking as in these examples, which reference the SEL-PROGRAM-MASK parameter.

SEL-PROGRAM-MASK Contents	Program Selected
PR*	Any program whose name begins with the two characters PR.
*HR	Any program whose last two characters are HR.
PR*HR	Any program whose name begins with the two characters PR and whose last two characters are HR.
HR??14*	Any program whose first two characters are HR, whose third and fourth characters are anything, and whose fifth and sixth characters are 14.

A field that supports masking has the word MASK as the last qualifier in the field name.

These are valid masking fields:

SEL-SUBSCHEMA-MASK	SEL-USERID-MASK
SEL-EXEMPT-SUBSCHEMA-MASK	SEL-TASK-NAME-MASK
SEL-PROGRAM-MASK	SEL-TERMINAL-MASK
SEL-PROGRAM-MASK	SEL-DDNAME-MASK
SEL-VERBNAME-MASK	

### **Record Occurrence Selection Criteria**

If, after applying the run unit selection criteria, Tracer has not excluded a run unit, Tracer applies the record selection criteria to determine if the record should be reformatted or ignored. All other selection criteria fields that are not previously listed as run unit selection criteria are considered record selection criteria.

When the IDMS journal record meets the stipulations defined here, Tracer sends the information in the reformatted file.

The criteria descriptions that follow list the selection parameters in alphabetical order, except where parameters are interdependent. When this is the case, Tracer groups the parameters.

To enable some of these selections, the CA-IDMS System Exit 14 must be installed. The definitions note those selection criteria that require CA-IDMS System Exit 14.

The DBTRFMT sample calling program allows overrides to the default for any of the selection criteria values. DBTRFMT first performs the code-in by the DEFAULTS copybook to set all values to the defaults specified for each selection parameter.

Next, DBTRFMT reads //DEFAULTS, the user-specified selection control card from the DBTINPUT file to override only those selection values that are changed.

**SEL-ABEND-SEQUENCE-NBR**

Use this parameter only if requested by ASG Customer Support for debug situations.

**SEL-DBKEY-LINE-INDEX**

Tells Tracer to only process record occurrences whose dbkey has this line index number. When used with the SEL-DBKEY-PAGE-NUMBER parameter, this allows the user to select only a specific dbkey. If the user knows the record type of the specified dbkey, Tracer runs faster if you use SEL-SINGLE-RECORDNAME along with this parameter.

Default = 0.

**SEL-DBKEY-PAGE-NBR-END-RANGE**

Use only when the SEL-YN-USE-RANGE-FOR-PAGE-NBR parameter is set to Y. This specifies the ending page number in a range of page numbers that Tracer uses to select only record occurrences whose page number portion of the dbkey matches this range.

Default = 0.

**SEL-DBKEY-PAGE-NUMBER**

Tracer processes only record occurrences whose dbkey has this page number. When the SEL-YN-USE-RANGE-FOR-PAGE-NBR parameter is set to Y, Tracer uses this parameter as the beginning number in the range. The default for this parameter is zero, which means to process without regard for the page number of the record's dbkey.

Default = 0.

**SEL-DDNAME-MASK**

Limits Tracer to processing updates to records contained in physical database files within the DMCL specified in the jobstreams described in ["Step 2 - DMCL and Schema Mapping" on page 21](#). Use only if SEL-YN-RETURN-DMCL-FIELDS = Y (see ["SEL-YN-RETURN-DMCL-FIELDS" on page 69](#)). Use ALLDDNMS to bypass this selection criterion.

Default = ALLVALUE.

**SEL-DEBUG-SEQUENCE-NBR**

Use this parameter only if requested by ASG Customer Support for debug situations.

### **SEL-DECOMPRESSION-DBPROCEDURE**

Specifies the name of the database procedure that carries out decompression if your site does compressions and decompressions but does not use IDMS COMP or PRESSPACK conventions. Change this parameter only if your site uses a decompression database procedure that is not provided by CA.

Default = \*NORMAL\*.

### **SEL-YN-CHECK-ELEM-VALUE-1-REC**

Setting this parameter to Y and using the parameter fields listed below, tells the DBTRACE Assembler language program to select occurrences of a specific record type only when a particular field in that record contains a particular value. This can be accomplished in a more user-friendly fashion with Tracer's SQL-like selection language, which works with the Tracker report. However, if you are only looking for one record type and one field value, it is more efficient to do it this way.

Default = N.

### **SEL-SINGLE-RECORDNAME**

Specifies the name of the record type to check.

Default = ALL-SUBS-RECORDS.

This feature has been enhanced to include/exclude multiple record names and even element names. See ["Including and Excluding Records and Elements" on page 74](#). When you want to include only one record type, SEL-SINGLE-RECORDNAME provides better performance.

### **SEL-YN-CHECK-ELEM-CHG-1-REC**

When set to Y, you must also specify the beginning offset and the length of the field that Tracer will look at to see if it has changed.

Default = N.

### **SEL-YN-CHECK-2ND-ELEM**

Selection criteria parameters whose names include both 2ND and ELEM are used only when the parameter SEL-YN-CHECK-ELEM-CHG-1-REC is set to Y. The purpose of these parameters is to allow a site to check whether a particular field has a particular value, even in circumstances where the SQL selection language technique cannot be used.

For instance, if a site has a record type defined in a schema that may have many different record layouts on the database, the application programs sometimes do not use the schema record layout at runtime to address various fields having the record occurrence. Rather, they may use one of any number of different record layouts based on the value in a TYPE field located in a header portion of the record occurrence.

The Tracker schema mapping load module (produced by jobstream @TB2BDM3) only has one record layout per record type and, for this reason, Tracer is unable to determine at run time that various different record layouts could be needed for the same IDMS schema defined record type. However, by using these parameters, a site can still selectively process only those record occurrences that have a certain value in the TYPE field in the first part of the record, and also a certain value in a particular field in the later redefined part of the record. [Figure 11](#) shows an example.

**Figure 11 • Selective Processing**

SEL-SINGLE-RECORDNAME	APP-PGM-DEF-REC
SEL-YN-CHECK-ELEM-VALUE-1-REC	Y
SEL-CHANGED-ELEM-OFFSET	3
SEL-CHANGED-ELEM-LENGTH	2
SEL-ELEM-VALUE-1-REC	05
SEL-YN-CHECK-2ND-ELEM	Y
SEL-2ND-ELEM-OFFSET	14
SEL-2ND-ELEM-LENGTH	9
SEL-ELEM-VALUE-2ND	IMPORTANT

The above parameter settings cause Tracer to only process (or select) updated record occurrences that are of the record type defined in the schema as APP-PGM-DEF-REC, and only those record occurrences where the value in the third and fourth character positions into the record occurrence contain the two digits 05, then also the positions 14 through 22 into the record occurrence must contain the character string *IMPORTANT*.

#### **SEL-2ND-ELEM-OFFSET**

Specifies the offset into the record occurrence where the field begins. This is determined by: the first position in the record should be denoted as 1, the second position as 2, and so on.

#### **SEL-2ND-ELEM-LENGTH**

Specifies the number of bytes required by the field.

#### **SEL-ELEM-VALUE-2ND**

Specifies the actual value that is being searched for in the field. If the field is not defined as DISPLAY or ZONED DECIMAL, then you must enter the actual value here using the hexadecimal capabilities of your editor (e.g., in TSO, use the HEX ON command). You must enter the value here exactly as it displays in the record occurrence. For instance, if the field you want to check is defined as PIC 9(5) and you are searching for the value 5, then you would enter 00005. However, if the field is defined as PIC 9(5) COMP-3, you would enter X'00005C.

**SEL-CHANGED-ELEM-OFFSET**

Specifies the offset where the field begins (Use IDMSRPTS RECDES report to determine).

Default = 0.

**SEL-CHANGED-ELEM-LENGTH**

Specifies the length of the field.

Default = 0.

**SEL-ELEM-VALUE-1-REC**

Indicates the value of the field to match. When this parameter is set to Y, Tracer examines each record occurrence of the SEL-SINGLE-RECORDNAME record. Tracer examines the specified offset for the specified length to determine whether the specified value is present. If it matches, the record occurrence is selected.

Default = N.

**SEL-EXEMPT-RECORDNAME**

Specifies record occurrences not to be processed by Tracer.

Default = NO-REC-TO-EXEMPT.

This feature has been enhanced to include/exclude multiple record names and even element names. See ["Including and Excluding Records and Elements" on page 74](#). When you simply want to exclude one record type, SEL-EXEMPT-RECORDNAME provides better performance.

**SEL-1ST-NBR-OF-RUN-UNITS**

Prompts Tracer to stop processing after it has processed all the records that meet the selection parameters for the indicated number of run units.

Default = 0.

**SEL-1ST-NBR-OF-TASK-IDS**

Prompts Tracer to stop processing after it has processed all the records that meet the selection parameters for the indicated number of tasks.

Default = 0.

**SEL-1ST-NBR-SELECTED-RECORDS**

Prompts Tracer to stop processing after it has processed the indicated number of record occurrences.

Default = 0.

**SEL-FV-FIXED-OR-VAR-OUTREC**

Allows you to build reformatted records using *occurs depending on* program logic. A record layout can also be used when accessing the data and/or pointers in the reformatted records without using an access program.

If this parameter is set to V, the relative offset in the reformatted records where the AFTR image of the data is located is different for each type of database record. The BFOR image requires varying amounts of room, depending on the record type. Programs that read the reformatted file use the *occurs depending on* field types in the reformatted record layout to obtain appropriate admissibility.

If you are using an ad hoc reporting tool and prefer not to use *occurs depending on* schemes, set this parameter to F. This ensures the areas in the reformatted record layout for RFJ-BFOR-POINTERS-AREA, RFJ-AFTR-POINTERS-AREA, RFJ-BFOR-DATA-AREA, and RFJ-AFTR-DATA-AREA always begin at a static offset in the record and each of the above fields is the same length for each record type.

When this parameter is set to F, the length of the pointer fields (RFJ-BFOR-POINTERS-AREA and RFJ-AFTR-POINTERS-AREA) is what you specify for the SEL-MAX-BYTES-FOR-POINTERS value.

The length of the RFJ-BFOR-DATA-AREA is what you specify for the SEL-MAX-BYTES-FOR-DATA value.

When this parameter is set to V, the length of the RFJ-AFTR-DATA-AREA depends on the record type. This keeps the length of the variable-length reformatted record as small as possible.

Default = V.

**SEL-HUND-SECS-WAIT-B4-COMT**

Defines the length of time run units can wait between updates before Tracer concludes they are waiting for another run unit to release a record lock.

Default = 201.

**SEL-LOCAL-OR-CV-JRNL**

Specifies the type of job that produced the IDMS journal archive tape you wish to reformat. These are the job types:

- C = Central Version
- L = Local Mode

Default = C.

**SEL-MAX-BYTES-FOR-DATA**

Determines the maximum number of bytes reserved for all the BFOR images written to the reformatted file when the SEL-YN-COMBINE-BFOR-AFTR-OUT parameter is set to Y. This value is ignored if SEL-YN-FIXED-OR-VAR-OUTREC is set to V.

Default = 1000.

**SEL-MAX-BYTES-FOR-POINTERS**

Controls two different decisions. If it is set to zero (0), Tracer does not write pointers to the reformatted file, even when the SEL-YN-RETURN-POINTERS parameter is set to Y.

This allows a single run of Tracer to produce lines in Report 01 for updates to pointers only. In the same run, a reformatted file is produced that does not contain records where only the pointer positions were updated.

If the SEL-YN-FIXED-OR-VAR-OUTREC parameter is set to F, this parameter controls the size of the portion of the reformatted record reserved to contain pointers (i.e., the RFJ-BFOR-POINTERS-AREA and/or the RFJ-AFTR-POINTERS-AREA).

When the SEL-YN-FIXED-OR-VAR-OUTREC is set to V, Tracer is not concerned with the actual value of this number, unless it is zero.

Default = 100.

**SEL-NAME-OF-ONLY-CHANGED-SET**

Tells Tracer to limit the records processed to only pointer-only updated records, where the set whose name is shown here was impacted by the database update.

Default = ALLVALUE.

**SEL-SELECT-PAGE-GROUP-NBR**

If non-zero, defines the page group associated with the subschema used by the application program that issued the update. Only those that match are processed. The SEL-YN-ONLY-1-PAGE-GROUP parameter must be set to Y for Tracer to use this parameter.

If Tracer is processing more than one segment (DBNAME) of a segmented database in a single run, and you use the SEL-DBKEY-PAGE-NUMBER parameter to limit processing by dbkey, this parameter may be needed. This is because it is possible for the same dbkey to exist in two different segments. In this case, the page number uniquely identifies a specific record occurrence.

Default = 0.

**SEL-SUBSCHEMA-NAME**

Only for earlier releases of Tracer. All information retrieved from the subschema load module is now retrieved from the DBTRECIN file created in ["Step 2 - DMCL and Schema Mapping" on page 21](#)).

**SEL-SCHEMA-NAME**

If you want to select records that were defined in only a single schema and you are pointing at a //DBTRECIN file that has many schemas defined in it, you may use this parameter to specify the schema you want.

**SEL-TYPE-UPDTS-USER-SYS-ALL**

Specifies a code telling Tracer what type of updates to process.

These are the valid values for this parameter:

Value	Description
U	= User requested updates. Select only records modified by an applications program.
S	= System requested updates. An area has been readied in update mode and IDMS has identified a fragmented record where one or more of the fragments may be brought back and combined with the root because there is now enough space on the target page. IDMS does this cleanup work even if there is no update verb issued by the applications program. This activity does not change the record occurrence logically. It only changes the physical location of parts of the record occurrence. When this happens, Tracer determines that the verb is AUTOUPDT.
A	= All records that were updated and journaled. When you specify A, all records that were on the journal are eligible for Tracer to process based on this parameter setting.

**SEL-UPDATE-TIME-START****SEL-UPDATE-TIME-STOP**

These UPDATE-TIME parameters allow the user to select only the updates that occurred during a specific time range. These parameters differ from SEL-RU-BEGIN-TIME and SEL-RU-END-TIME because the latter refers to the time the updating run unit started. These parameters are particularly useful if the user wants to see some of the updates done in the middle or toward the end of a long-running update job.

**SEL-USER-BLT-RUAL-TYPE-PROG**

Tells Tracer the name or naming convention of your RHDCRUAL user-written, long-running system overhead programs. Normally, this parameter is blank because only a small number of IDMS sites have this type of program running in their IDMS CV.

Default = spaces.

### **SEL-VERBNAME-MASK**

Limits Tracer processing to certain verbs, some of which have qualifiers. For example:

```
ER*  ERASE ALL
      ERASE PRM
      ERASE SEL
```

Use ALLVERBS to bypass this selection criteria.

Default = ALLVALUE.

### **SEL-YN-ABEND-MODE**

Use this parameter only if requested by ASG Customer Support for debug situations.

### **SEL-YN-ALWAYS-SEL-DBOL-DMLO**

Indicates whether Tracer will try to select updates performed by run units for either the DBOL or DMLO programs. If this parameter is set to Y, updates done by DBOL or DMLO for a given subschema associated with the schema, specified in ["Step 2 - DMCL and Schema Mapping" on page 21](#)), display on Tracer Report 01.

Since these programs enable the online user to update any database record without writing an applications program, auditors often are particularly interested in tracking them. If you set this parameter to Y, Tracer does not de-select any DBOL or DMLO run units based on the run unit selection criteria, even if the DBOL or DMLO run unit would not normally meet that criteria.

Default = N.

### **SEL-YN-COMBINE-BFOR-AFTR-OUT**

Specifying Y allows Tracer to combine, as a single record (or a single line in Report 01), the BFOR and AFTR IDMS journal image information associated with the DML verb that changed an existing database record occurrence.

Specifying N writes two separate records to the reformatted file (or two lines to Report 01). One record contains information about the appearance of the database record before it was changed (from the BFOR journal record). The other contains information about the appearance of the database record after it was changed (from the AFTR journal record).

Default = Y.

### **SEL-YN-DBKEY-IN-ANY-PNTR-POS**

If this parameter is set to Y and the SEL-DBKEY-PAGE-NUMBER parameter is non-zero, Tracer processes any record occurrence that would have been selected based on other parameters only if any of the dbkeys in its pointer positions match the specified page number or range of page numbers.

Default = N.

**SEL-YN-DBKEY-IN-CHGD-PTR-POS**

If this parameter is set to Y and the SEL-DBKEY-PAGE-NUMBER parameter is non-zero, Tracer processes any record occurrence that would have been selected based on other parameters only if any of the dbkeys in one of the changed pointer positions match the specified page number or range of page numbers and were changed to another dbkey or set to Null.

Default = N.

**SEL-YN-DBOL-DMLO-IN-BOTH-FILES**

Indicates whether Tracer should write records that were updated to both the DBTRFMTJ and the DBOLDMLO file if they were updated either by DBOL or DMLO and if the SEL-YN-WRITE-DBOLDMLO-FILE parameter was set to Y.

Default = Y.

**SEL-YN-DBT-WRITE-REFORMATTED**

Specifying Y writes any records Tracer processes (based on selection criteria) to the reformatted file (DBTRFMTJ).

If you want your program code to determine which records are written to the reformatted file, set this parameter to N. If your program code determines that a particular record should be written to the reformatted file, then you would call Tracer again with the TR-I-W-WRITE-THIS-RFMT-REC flag set to W.

Whenever Tracer determines that a record should be processed (i.e., the TR-O-YN-RECORD-READY flag is set to Y), it checks two input fields before writing the reformatted record. If TR-I-YN-DBT-WRITE-REFORMATTED is Y, then Tracer writes the record. If N, and if TR-I-W-WRITE-THIS-RFMT-REC is W, Tracer writes the reformatted record.

Default = Y.

**SEL-YN-DBTRUNUI-BGIN-COMT**

Indicates whether both journal records (BGIN and COMT) are to be written to the DBTRUNUI file in addition to ENDJ and ABRT records (which are always written when the SEL-YN-WRITE-DBTRUNUI parameter is set to Y).

Default = N.

### **SEL-YN-GET-SR2-FROM-VSAM**

This parameter is applicable only if you have relocated records on your physical database. Relocated records are sometimes produced by the Restructure utility when you make a restructured record occurrence larger and there is not enough room for the entire record occurrence on the physical page where it is stored. IDMS puts a record stub (record ID of SR2) on the original page and stores the main part of the record occurrence on another page. A record ID of SR3 is associated with the record occurrence that is stored in the new location.

When Tracer encounters an SR3 record ID on the journal, Tracer is able to determine the record's type, even if this parameter is set to N. If you set this parameter to Y, Tracer takes the dbkey of the SR2 record that points at this record occurrence and returns it to the reports and/or the reformatted file. If you set this parameter to Y, you must have already run the jobstreams whose names start with @R for this particular IDMS-CV environment. These relocated setup jobstreams produce a VSAM file that is keyed by the dbkey and page group of the SR3 dbkey and has the dbkey of the associated SR2 record occurrence and its original record ID value. When you set this parameter to Y, you must also have in your jobstream the //VSAMIN DD statement. Here, you specify the VSAM file that was built earlier. Tracer then accesses this VSAM file whenever it encounters a relocated SR3 type record occurrence on your journal tape for a record occurrence in a selected run unit.

### **SEL-YN-DE-RELOCATE-IF-NEEDED**

Indicates if records that were relocated when a restructure was done should be processed by Tracer. Relocated records are represented on the IDMS database by a system record ID of SR3. Parameters whose first characters are SEL-RESTRUCT need to be specified when this parameter is set to Y and the user suspects that there may be SR3 records journaled.

Default = Y.

### **SEL-YN-DECOMPRESS-IF-NEEDED**

Indicates that compressed records should be processed by Tracer.

Default = Y.

### **SEL-YN-DEFRAGMENT-IF-NEEDED**

Indicates that fragmented records should be processed by Tracer.

Default = Y.

### **SEL-YN-DESPAN-IF-NEEDED**

Indicates that spanned records should be processed by Tracer.

Default = Y.

**SEL-YN-GLOBAL-SUBSCHEMA**

Use this parameter only if System Exit 14 is not applied at your site. If System Exit 14 is applied, use the SEL-SUBSCHEMA-MASK parameter.

Allows you to reduce the amount of CPU cycles that Tracer requires to analyze record selection criteria for run units that passed the run unit selection criteria options. If the DBTRECIN file created in ["Step 2 - DMCL and Schema Mapping" on page 21](#) contains all the records defined in the selected schema, specify Y.

When Tracer encounters a record type that is not defined in the DBTRECIN file, Tracer knows there is no need to check record selection criteria options for any subsequent records updated by this run unit.

Default = N.

**SEL-YN-LIMIT-TO-PAGE-GROUP-C**

Indicates whether Tracer considers only concurrent run units that are associated with the same page group as the run unit that waited for a dbkey lock when determining what caused bad response time.

Default = Y.

**SEL-YN-ONLY-1-PAGE-GROUP**

Indicates whether Tracer should limit the record occurrences processed to only those that were updated by a subschema that was associated with a specific page group number. When this is set to Y the SEL-SELECT-PAGE-GROUP-NBR parameter must be specified.

Default = N.

**SEL-YN-ONLY-TARGET-OF-DML-RC**

Indicates whether Tracer should select only those records that were the target of the updating DML verb. This parameter is applicable to the production of the reformatted journal archive file (//DBTRFMTJ).

Default = N.

### **SEL-YN-TARGET-PTRO-NOT-CON-DI**

Tells Tracer whether it should process journaled records that are for the TARGET-OF-DML-VERB but contain only changes to the pointer portion of the record when the update verb is not CONNECT or DISCONNECT. The TARGET-OF-DML-VERB refers to the record occurrence that the application program was referring to when the update verb was issued (e.g., on a STORE verb, it is the record occurrence that is to be stored, as opposed to all the record occurrences for which IDMS must adjust pointers in sets in which the newly stored record occurrence participates). In some cases, IDMS will journal the pointer only portion of the target record first and then later journal the pointers and the data of that record (as in the case of a MODIFY verb where one of the sort key fields were changed—IDMS must readjust the position of the changed record occurrence in the sorted set(s) where it participates based on the new value of the sort key. When doing this, IDMS journals the pointer only part of the target record occurrence before journaling the data portion).

When this parameter is set to N, then only the record occurrence that has the data part of the modified record occurrence is processed. This parameter is set to Y only if you need to see all record occurrences that were journaled (usually this is done only for performance analysis reasons).

### **SEL-YN-PROCESS-ENDEVOR-RUS**

Indicates whether Tracer should exempt Endeavor long running system run units from normal processing. This parameter is only applicable to those shops that use Endeavor. This parameter is normally set to N since Endeavor's BNVPSEV program is functionally very similar to RHDCRUAL, in that it runs all day and services requests from tasks associated with a variety of different users/tasks/terminals.

Default = N.

### **SEL-YN-PROCESS-RHDCRUAL-RUS**

Indicates whether Tracer should treat RHDCRUAL run units like any other program. This parameter is normally set to N because RHDCRUAL programs only update the dictionary and cannot be traced to individual user IDs, because these run units are active throughout the entire IDMS CV run.

Default = N.

### **SEL-YN-PROCESS-USER-RUAL-TYP**

Informs Tracer whether you have a site-written program similar to the RHDCRUAL program and you want to exempt it from normal processing consider-actions. This parameter is normally set to N.

If you have a program that runs in the background all day and performs updates based on calls from other user-oriented tasks, you may not want Tracer to process these updates, as you cannot be sure which user was responsible for calling the overhead program.

Default = N.

**SEL-YN-RETURN-DMCL-FIELDS**

Allows Tracer to bypass looking up DMCL information such as ddname, schema, and schema version. This speeds up run time for those who do not use these parameters.

Default = Y.

**SEL-YN-RETURN-ONLY-CHNGD-MOD**

Specifying Y bypasses record updates written to the IDMS journal but not actually changed by a MODIFY verb. Specifying N reports all indications of record updates in the reformatted file.

Default =Y.

**SEL-YN-RETURN-POINTERS**

Specifying Y includes information about updates to pointer positions on Tracer reports and in the reformatted file.

If this parameter is set to Y and SEL-MAX-BYTES- FOR-POINTERS = 0, the reports indicate those records that had their pointers updated and the reformatted file only contains records whose data was updated.

Default = Y.

**SEL-YN-RETURN-SYSTEM-RECORDS**

Specifying Y processes IDMS system overhead records (e.g., SR7, SR8 used for indexing) for selected run units. This information then displays on Report 01.

Default = Y.

**SEL-YN-USE-RANGE-FOR-PAGE-NBR**

Indicates whether Tracer should look for a range of page numbers or just use the page number specified on the SEL-DBKEY-PAGE-NUMBER parameter. When you set this parameter to Y, you should specify the SEL-DBKEY-PAGE-NBR-END-RANGE parameter to indicate the ending page number in the range and you use the SEL-DBKEY-PAGE-NUMBER as the beginning page number.

Default = N.

See related parameters:

- ["SEL-DBKEY-LINE-INDEX" on page 57](#)
- ["SEL-DBKEY-PAGE-NBR-END-RANGE" on page 57](#)
- ["SEL-DBKEY-PAGE-NUMBER" on page 57](#)
- ["SEL-SELECT-PAGE-GROUP-NBR" on page 62](#)
- ["SEL-YN-DBKEY-IN-ANY-PNTR-POS" on page 64](#)
- ["SEL-YN-DBKEY-IN-CHGD-PTR-POS" on page 65](#)
- ["SEL-YN-ONLY-1-PAGE-GROUP" on page 67](#)

**SEL-YN-WRITE-DBOLDMLO-FILE**

Writes each selected record updated by DBOL or DMLO to the DBOLDMLO file. Most IDMS shops want to track these types of updates regardless of other things they may be doing with Tracer. This gives you the ability to always produce a reformatted file that exclusively contains updates done by DBOL or DMLO users, in addition to the normal reformatted file.

Default = Y.

**SEL-YN-WRITE-DBTRUNUI**

Allows you to produce the DBTRUNUI file which contains run unit information.

Default =Y.

**SEL-YN-WRITE-SYSTEM-RECS-RFMT**

Specifying Y includes system update records on the reformatted file.

Default = N.

**SEL-YN-WRITE-USER-DBSTAMP**

Produces the DBSTAMP file, which contains information written to the journal file by IDMS System Exit 14.

Default = Y.

**YN-PRINT-NON-TARGET**

Instructs Tracer whether or not to print records if they were not the target of the DML verb. This parameter controls only the printing of these records, not the creation of the reformatted file.

Synonym: YN-PRINT-NON-TARGET-RECS

These are the default Tracer selection criteria:

Selection Name	Default
SEL-SUBSCHEMA	IDMSNWKA
SEL-MAX-BYTES-FOR-DATA	0000000500
SEL-MAX-BYTES-FOR-POINTERS	0000000000
SEL-RU-ENVIRONMENT	DBDC
SEL-YN-RETURN-POINTERS	N
SEL-YN-COMBINE-BFOR-AFTR-OUT	Y

---

Selection Name	Default
SEL-DECOMPRESSION-DBPROCEDURE	*NORMAL*
SEL-FV-FIXED-OR-VAR-OUTREC	V
SEL-YN-RETURN-SYSTEM-RECORDS	N
SEL-YN-WRITE-SYSTEM-RECS-RFMT	N
SEL-YN-RETURN-ONLY-CHGD-4-MOD	N
SEL-YN-DBT-WRITE-REFORMATTED	Y
SEL-RU-BEGIN-DATE	ALLVALUE
SEL-RU-END-DATE	MM/DD/YY
SEL-RU-BEGIN-TIME	0000000000
SEL-RU-END-TIME	0000000000
SEL-YN-GLOBAL-SUBSCHEMA	N
SEL-LOCAL-OR-CV-JRNL	C
SEL-PROGRAM-MASK	ALLVALUE
SEL-VERBNAME-MASK	ALLVALUE
SEL-DDNAME-MASK	ALLVALUE
SEL-TASK-NAME-MASK	ALLVALUE
SEL-TERMINAL-MASK	ALLVALUE
SEL-USERID-MASK	ALLVALUE
SEL-SUBSCHEMA-MASK	ALLVALUE
SEL-EXEMPT-SUBSCHEMA-MASK	NOEXEMPT
SEL-TYPE-RUN-UNIT-SELECTION	A
SEL-RUN-UNIT-NUMBER-1	0000000000
SEL-RUN-UNIT-NUMBER-2	0000000000

Selection Name	Default
SEL-TYPE-TASK-ID-SELECTION	A
SEL-TASK-ID-NUMBER-1	0000000000
SEL-TASK-ID-NUMBER-2	0000000000
SEL-SINGLE-RECORDNAME	ALL-SUBS-RECORDS
SEL-YN-CHECK-ELEM-CHG-1-REC	N
SEL-CHANGED-ELEM-OFFSET	0000000000
SEL-CHANGED-ELEM-LENGTH	0000000000

---

## Default Input File (Tracer) Parameters

YN-USE-DEFAULT-FILE allows you to create a site-defined or a user-defined default file without having to make program changes.

For example, the default for most of the advanced options is off. You could turn these features on by creating a site-defined default parameter file. You would then update your standard Tracer JCL or PROC to point to this dataset.

To implement this feature, decide what your site-defined defaults should be. Enter the necessary parameters into a PDS member or sequential file (possibly stored as a control library).

Add this line of JCL to your PROC or jobstream:

```
//DEFAULTS DD DSN=xxxx.yyy.zzzz(member),DISP=SHR
```

Add this parameter as the first line of any user-submitted job:

```
//DBTINPUT DD *  
YN-USE-DEFAULTS-FILE Y
```

This parameter line and value instructs Tracer to switch from reading the DBTINPUT file to reading all the parameters in the default file.

When an end-of-file is found on the default file, Tracer then returns and reads the remainder of the DBTINPUT file (from whatever point it left off). This is why it is important to include this parameter as the first parameter in the DBTINPUT file. Then, any parameter settings found in the DBTINPUT file override the settings previously set from the default file.

This feature may also be used to shield end-users or the reports from some of the more esoteric Tracer parameters.

## Field-level Selection Parameters

The user may limit the records printed on the Tracker report by using SQL-like selection criteria. [Figure 12](#) shows an example.

**Figure 12 • Limiting Tracker Report Records**

```
//DBTINPUT DD *
TRKR-YN-USE-FLDSELS                Y
//FLDSELS DD *
  SELECT ESALHIST-8002 WHERE DELTA% SALARY-AMT > 10
                                OR DELTA SALARY-AMT > 450.00
SELECT EMPLOYEE-8001 WHERE
  EITHER LAST-NAME LIKE 'JOHN%'
    AND (LATEST FIRST-NAME = 'SAM' OR
         LATEST CITY = 'MIAMI')
  OR DBKEY = 24681:13 OR USERID = XYZ001
```

Another feature is the ability to specify (limit) the names of elements you wish to see on the Tracker report (and the ability to include or exclude multiple records or elements), as depicted in [Figure 13](#).

**Figure 13 • Limiting Tracker Report Data**

```
//DBTINPUT DD *
TRKR-YN-USE-FLDINCL                Y
//FLDINCL DD *
EXC ALL
INC SR-036
EXC SR-036                DATE-LU-036
EXC SR-036                PREP-BY-036
INC INQ-058
```

A third feature is the ability to print the Tracker report in 80-column mode. This allows you to see most of the Tracker report on an 80-column display device. When this feature is turned on, the before/after images are printed on top of each other, rather than left/right of each other, thus making the report longer but easier to read on a terminal.

## Field-level Selection

These are the ways to use Field-level Selection

- To list only salary records where salary amount is more than 15% greater than the old salary amount.
- To list only changes made by a specified user ID.
- To list only changes made to a specific department or part-number.
- To list only changes made to EMPLOYEE records where EMPLOYEE-NAME contains the value SMITH.
- To select the CALC-KEY.

For example, a record occurrence is suddenly missing from the database. You want to know who deleted it and the information that the record contained so that you can re-enter it using DBOL or DMLO. Use Tracker's new field-level selection criteria to select the CALC-KEY of the before record image.

- To investigate what caused a broken chain.

For example, You run a program against the journal to find when a certain pointer changed, and who or what caused the change.

## Including and Excluding Records and Elements

A separate ddname (//FLDINCL) can contain user input specifying which records and/or elements to include or exclude from the Tracker report. The main purposes of this feature are to make the report more readable and to shorten the amount of printed output. This feature is only activated when the parameter TRKR-YN-USE-FLDINCL is set to Y.

These are the rules for coding input:

Column	Rule
1	May contain * or \$ for a comment
2-4	Must contain EXC for exclude, or INC for include
6-37	Contain a record name or the word ALL
39-61	Contain an element name or the word ALL

---

## Example 1:

Include only a single record type on the Tracker report, but exclude three specific elements from being printed. [Figure 14](#) shows an example.

**Figure 14 • Include Single Record Type Only**

```
//FLDINCL DD *
EXC ALL
INC INQ-058
EXC INQ-058DATE-LU-058
EXC INQ-058TIME-LU-058
EXC INQ-058REV-BY-058
```

## Example 2:

Print all records encountered on the journal (for the Tracker subschema) except for three specific record types. [Figure 15](#) shows an example.

**Figure 15 • Exclude Specific Record Types**

```
//FLDINCL DD *
INC ALL
EXC RCDSYN-079
EXC OOK-012
EXC TIME-LU-058
```

## Example 3:

List only three specific elements to be included on the report. [Figure 16](#) shows an example.

**Figure 16 • Include Specific Record Types Only**

```
//FLDINCL DD *
EXC ALL
INC INQ-058INQ-NAM-058
INC INQ-058REV-BY-058
INC INQ-058DATE-LU-058
```

When Tracker encounters an element on the journal, it looks to see if that element-name was specifically included or excluded. If no specific include or exclude for the element was found, Tracker looks for a specific option for the record name. If no such record name is found in an include or exclude list, then the global option EXC ALL or INC ALL is applied.

**Note:**

Elements that are part of the control key are always printed if the other fields on the record are printed (i.e., any EXC for control-key elements would not take place).

## SQL Selection Parameters

This section provides information on:

- Syntax requirements
- SQL additions for before and after images
- Selection criteria

You use the SQL selection criteria only when this parameter is set to Y:

```
//DBTINPUT DD *  
TRKR-YN-USE-FLDSELS  Y
```

### Syntax Requirements

Syntax requirements for SQL selection parameters are: free format input; single or double quotes; allowable logical operators, and supported SQL predicates. Each of these is explained in this section.

#### Free Format Input

The user input is completely free format, with the exception that keywords and variables must be separated by spaces. Comparison operators and parenthesis do not require space separation.

Example:

```
//FLDSELS DD *  
SELECT EMPLOYEE WHERE (BEFORE LAST-NAME='SMITH')  
AND BEFORE CITY = 'MIAMI'
```

You may code one and only one SELECT statement for each IDMS record type. Each SELECT statement must begin on a new line. The SELECT keyword must begin in columns 1 through 3.

When the SQL-like statement is parsed, a token listing is written to the SYSOUT, which numbers the tokens and displays each token individually. If you have a syntax error, the error refers to a token (or word) number, leaving no doubt as to where the error is located. The number to the right indicates the length (in bytes) of the token, and the next code indicates the type of token. [Figure 17](#) is an example:

**Figure 17 • Syntax Error Sample**

```

1 SELEC          SIZE=0007 TYPE=N
2 EMPLOYEE      SIZE=0008 TYPE=N
3 WHERE         SIZE=0005 TYPE=N
4 (             SIZE=0001 TYPE=N
5 BEFORE        SIZE=0006 TYPE=N
6 SALARY-AMT    SIZE=0009 TYPE=N
7 >             SIZE=0001 TYPE=N
8 100.00        SIZE=0005 TYPE=Y
9 )             SIZE=0001 TYPE=N
ETC...
```

These are the valid type codes:

Code	Description
Y	= Numeric
N	= Not-numeric (alpha)
Q	= Quoted (alpha in quotes)
(	= Compound token (subscripted or function)

### Single and Double Quotes

You may use either single or double quotes. These quotes are optional unless there are embedded blanks within one word. For example:

```
CITY = 'DAYTONA BEACH'
```

**Or**

```
CITY = NAPLES
```

If the quotes were left off, the program would look for CITY = DAYTONA and then issue a syntax warning about the word BEACH, expecting to find a parentheses or a key word such as AND or OR.

### Allowable Logical Operators

Allowable logical operators in the WHERE clause include: AND, OR, NOT, and left and right parentheses.

The keyword NOT may be used in this manner:

```
SELECT EMPLOYEE-8001
WHERE NOT AFTER LAST-NAME LIKE 'JOHN%'
```

However, do not use the word NOT as part of a comparison operator.

For example, instead of coding:

```
SELECT EMPLOYEE-8001
WHERE AFTER LAST-NAME NOT = DOE
```

You could code any of these:

```
SELECT EMPLOYEE-8001
WHERE AFTER LAST-NAME NE DOE
```

```
SELECT EMPLOYEE-8001
WHERE AFTER LAST-NAME <> DOE
```

```
SELECT EMPLOYEE-8001
WHERE NOT AFTER LAST-NAME = DOE
```

Complex combinations of the variables can be used, as seen in [Figure 18](#).

**Figure 18 • Complex Variable Combinations**

```
SELECT EMPLOYEE-8001 WHERE (
((VERB = MODIFY AND
(EITHER STATE EQ 'FL' OR EITHER STATE EQ 'GA'))
OR
VERB = STORE AND
(AFTER STATE NE 'TX' AND AFTER STATE NE 'OK'))
AND
LATEST LAST-NAME LIKE 'JOHN%'
AND LATEST LAST-NAME <> 'JOHNSTON'
) OR VERB = ERASE
```

The previous example selects all employees whose last name starts with JOHN (except JOHNSTON), prints the journal record when someone moves in or out of Florida or Georgia or when someone is added to a state other than Texas or Oklahoma, and also shows all erased employee records (regardless of name).

### Supported SQL Predicates

SQL predicates supported by Tracer are LIKE, BETWEEN, and IN.

For example:

```
SELECT EMPLOYEE-8001 WHERE
      EITHER LAST-NAME-8001 LIKE 'JOHN%'
      AND AFTER EMP-STATE-8001 IN (FL,OK,TX,WY)
```

```
SELECT ESALHIST-8002 WHERE
      AFTER SALARY-AMT BETWEEN 1000 AND 1200
```

**Note:** \_\_\_\_\_

For the IN predicate, there is a limit of 20 values inside the parentheses.

---

### Non-supported SQL Predicates

SQL predicates that are not supported by Tracer at this time are NULL, ALL, SOME, ANY, EXIST and aggregate functions.

NULL checks for blanks and zeroes.

Example:

```
WHERE AFTER SALARY-AMT IS NULL
```

**Note:** \_\_\_\_\_

Since there is not a true NULL value in a network IDMS database, as there is in a TRUE SQL database, you may code:

```
WHERE AFTER SALARY-AMT = 0
```

**Or**

```
WHERE AFTER LAST-NAME = ' '
```

---

Predicates that require a subselect statement are not available in Tracer (they refer to matching values in another SQL table or list). These predicates are: ALL, SOME, ANY, EXIST.

Aggregate functions (such as MIN, MAX, AVG, COUNT, SUM) are *not* supported.

## SQL Comparison Operators

These standard comparison operators can be used to compare alpha to alpha or numeric to numeric:

EQ	LT	LE	GT	GE	NE
=	<	<=	>	>=	<>

## Functions

There are two types of complex tokens handled by Tracer: function calls and subscripted variables.

This is a sample built-in function:

```
SELECT EMPLOYEE WHERE SUBSTR(EITHER HIRE-DATE, 7, 2) = '93'
```

Tracer includes these built-in functions:

```
SUBSTR  
NAMELIKE  
IS-CHANGED
```

## SUBSTR Function

The SUBSTR (substring) function allows you to test the values of several bytes anywhere inside of an alphabetic field. In the example above, the variable HIRE-DATE is being tested. The number 7 indicates the character where SUBSTR starts, and the number 2 indicates the length of the comparison. For example, if HIRE-DATE were stored as 03/01/92, you could use the above SUBSTR function to test just the year, since year starts in byte seven and has a length of two.

## Example User-written Function

```
SELECT EMPLOYEE WHERE  
GREGDATE (LATEST HIRE-DATE, 'YY/MM/DD') < '08/24/01'
```

For this example, Tracer would show you all employee records updated on the journal for employees that were hired before February 1, 1991.

Suppose HIRE-DATE is stored as a Julian date on the database, in the *YYDDD* format, but your auditors always wish to test the date in the Gregorian format, *YY/MM/DD* (with the year first so that the greater than and less than symbols work appropriately). When Tracer detects the above syntax, it first tests to see if GREGDATE is a valid variable name in your schema.

If not, Tracer treats GREGDATE as a program name, and calls it, passing each of the parameters inside the parentheses as arguments to the GREGDATE subroutine.

If the program is not found in the load libraries in the //STEPLIB concatenation, your job will abend and Tracer will issue this standard MVS message:

```
S806 - REQUESTED MODULE NOT FOUND
```

[Figure 19](#) displays the standard calling format for user-written subroutines.

**Figure 19 • Calling Format for User-written Subroutines**

```
LINKAGE SECTION.
  01 DBCUCOMM-BLOCK.
    05 DBCUCOMM-RETURN-ALPHA PIC X(80).
    05 DBCUCOMM-RETURN-NUMBER PIC S9(11)V9(7) COMP-3.
    05 DBCUCOMM-RETURN-TRUE-FALSE PIC X.
    05 DBCUCOMM-ALPHA-PARM PIC X(80) OCCURS 8.
    05 DBCUCOMM-NUMERIC-PARM PIC S9(11)V9(7) COMP-3
                                     OCCURS 8.
    05 DBCUCOMM-FUTURE PIC X(500).

PROCEDURE DIVISION USING DBCUCOMM-BLOCK.
```

A sample COBOL user-written function is included with Tracer. It is called DBFUNCTN. It demonstrates how you might code a general purpose date conversion routine. The first parameter is the date to be converted, the second parameter tells the subroutine the expected format of the date to be converted, and third parameter tells the subroutine the format of the output date.

### Example User-written Function

```
SELECT EMPLOYEE WHERE
      DBFUNCTN(LATEST HIRE-DATE, 'MMDDYY', 'YY/MM/DD') <'08/24/01'
```

For this example, Tracer would call program DBFUNCTN. Suppose HIRE-DATE has a database picture of X(6) and is stored in the MMDDYY format. Suppose that the value of LATEST HIRE-DATE for our current journal record is 021201.

These parameters would be passed from the current journal record:

```
ALPHA-PARM (1) = 021201    the LATEST HIRE-DATE
```

This is how the program would interpret the parameters internally:

```
ALPHA-PARM (2) = MM/DD/YY
ALPHA-PARM (3) = YY/MM/DD
NUMERIC-PARM (1) = 021201    since date is all numeric
NUMERIC-PARM (2) = 0
NUMERIC-PARM (3) = 0
```

If any of the parameters or variables are all numeric, then Tracer passes them in both ALPHA-PARM (n) and the corresponding NUMERIC-PARM (n).

Single and double quotes around alpha fields are removed before placing them in the DBCUCOMM-ALPHA-PARM array.

Your program may return one and only one value in either the field DBCUCOMM-RETURN-ALPHA or DBCUCOMM-RETURN-NUMBER.

Thus, after Tracer has called the user-written subroutine, the select is simplified to:

```
SELECT EMPLOYEE WHERE '01/02/12' < '00/05/21'
```

which is FALSE and causes the record not to be selected.

Of course, your program's load module must be available in one of the datasets concatenated in the STEPLIB to avoid an S806 abend.

If you use the word EITHER or BOTH in front of a variable name, and the verb of the journal record is MODIFY, your function is called twice, once with the BEFORE value and once with the AFTER value. The EITHER/BOTH logic is applied. This may cause unpredictable results when your subroutine uses multiple variables.

Thus, if your user-written function requires multiple variables, instruct your users not to use the EITHER or BOTH keyword, but limit them to AFTER, BEFORE, EARLIEST, LATEST, DELTA, and DELTA%, and instruct them to code an AND/OR and repeat the entire function call, such as:

```
SELECT EMPLOYEE WHERE
      DBFUNCTN (BEFORE
      HIRE-DATE, 'MM/DD/YY', 'YY/MM/DD') < '00/05/21'
OR
      DBFUNCTN (AFTER
      HIRE-DATE, 'MM/DD/YY', 'YY/MM/DD') < '00/05/21'
```

Example subscripted variable:

```
SELECT EMPLOYEE WHERE
      EITHER SKILL-8003 (1,2,3) = '01'
```

## **NAMELIKE Function**

The NAMELIKE function is a special user-written function that is useful for comparing program names, terminal IDs, etc. It is similar to the SQL LIKE command, but provides a few additional features.

The primary difference is that the NAMELIKE function can be used to match program names that start or end with a certain phrase. This routine assumes that there are no embedded blanks in the field being matched (e.g., program names may never contain a blank other than at the end of the eight-character program name). You can never have more than one % sign in the NAMELIKE mask.

In summary, NAMELIKE is useful for testing short eight-character names, whereas the SQL LIKE is better for testing lengthy alpha fields or for accomplishing the IDMS CONTAINS logic.

This table compares the NAMELIKE function to the SQL LIKE function. Notes following the table explain some of the situations.

Mask	TEST-FIELD	NAMELIK E Result	SQL LIKE Result	Note
HR%SS	HR0020SS	T	T	
HR%SS	HR0020SSXYZ	F	F	1
HR%SS	HRXYZSS	F		2
%SS	ABCSS	T	T	
%SS	ABCSSXYZ	F	T	1
AB%	ABCSS	T	T	
AB%	ABCSSXYZ	T	T	
%AB%	WFABXYZ	T	T	3
%AB%	WFQQXYZ	T	T	3

**Note 1:** The NAMELIKE function returns a False because the TEST-FIELD does not end with the letters SS.

**Note 2:** The NAMELIKE function returns a value of False. Since NAMELIKE is a function, it returns a value (T or F), which must be tested. For example:

```
SELECT EMPLOYEE-8001 WHERE NAMELIKE (PROGRAM, 'HR%SS')='T'
```

```
SELECT EMPLOYEE-8001 WHERE PROGRAM LIKE 'HR%SS'
```

**Note 3:** %AB% is the same as the IDMS CONTAINS logic (e.g., WHERE LAST-NAME CONTAINS %AB%). The NAMELIKE function always returns a TRUE value when the mask field starts and ends with a % sign.

## IS-CHANGED Function

This statement causes Tracer to select EMPOSITION records only when the before or after images of the field SALARY-AMOUNT-0420 is changed when the update verb is a MODIFY.

**Note:** \_\_\_\_\_

It doesn't matter how the picture type of the field was defined.

---

```
SELECT EMPOSITION WHERE SALARY-AMOUNT-0420 IS-CHANGED AND VERB=MODIFY
```

It is important to use the VERB=MODIFY parameter to qualify the IS-CHANGED selection because other verbs like STORE and ERASE always indicate that the before or after images have changed.

The example in [Figure 20](#) demonstrates all the control parameters you would use to tell Tracer to only select the EMPLOYEE record occurrences where the EMP-LAST-NAME-0415 field was changed in any way by a MODIFY verb:

**Figure 20 • IS-CHANGED Function Example**

---

```
//DBTINPUT DD *
PRINT REPORT TRKR                                Y
TRACKER-SCHEMA-NAME                             EMPSCHM
SEL-SINGLE-RECORDNAME                            EMPLOYEE
TRKR-YN-PRINT-RECS-NOSELECT                     N
TRKR-YN-USE-FLDSELS                             Y
/*
//FLDSELS DD *
SELECT EMPLOYEE WHERE BEFORE EMP-LAST-NAME-0415 IS-CHANGED
AND VERB = MODIFY
*
```

---

## Non-database Dependent Fields

Every journal record has other key information that also displays on the Tracker report. You may also select record types by using these variables:

- Dependent on implementation of the Tracer-provided IDMS User Exit 14:  
USERID, SUBSCHEMA, TERMINAL, TASKNAME
- Available regardless of User Exit 14:  
VERB, DBKEY, PROGRAM

Examples:

```
SELECT ESALHIST-8002 WHERE USERID <> 'SHARON'
      OR NOT TERMINAL LIKE 'VTAM3%'
```

Selects all journal records for the ESALHIST-8002 record where the updates were not made by the user ID SHARON or were made by any terminal not in the payroll department. This would include updates done by user ID SHARON on any terminal where the first five characters were something other than VTAM3.

```
SELECT ESALHIST-8002 WHERE TASKNAME = 'DMLO'
      OR TASKNAME = 'DBOL'
```

Finds out what changes were made to this record-type by anyone using DMLO or DBOL.

```
SELECT ESALHIST-8002 WHERE TASKNAME <> 'PR'
      OR NOT NAMELIKE (PROGRAM, 'PR%C') = 'T'
```

Finds changes made to this record type by anyone not using the standard PR task or not using one of the standard PR programs.

```
SELECT ESALHIST-8002 WHERE VERB = 'STORE'
```

Limits the Tracker report to records that were stored.

```
SELECT ESALHIST-8002 WHERE DBKEY = 12001:045
SELECT EMPLOYEE-8001 WHERE
      DBKEY BETWEEN 12001:000 AND 12001:999
```

Limits the Tracker report to salary history records with the above dbkeys or dbkey ranges.

### *Limits of SQL Implementation*

- There is a limit of 80 tokens per SELECT statement. A token includes each parenthesis, variable (excluding the BEFORE and AFTER type qualifiers), comparison operator, and keyword such as SELECT and WHERE.
- There is a limit of five SELECT statements per run.
- There is a limit of 20 complex tokens per SELECT statement. (A complex token is a token that must be broken down into sub-tokens, which include subscripted variables, built-in functions, and user-written functions.)
- There is a limit of 20 values inside the parentheses of an IN list, such as:

```
WHERE AFTER STATE IN (FL, OK, TX, WY)
```

This example has four values inside the parentheses.

- There is a limit of 300 record or element names in the //FLDINCL file.
- Mathematical statements are not supported in the Tracer's SQL-like SELECT statement. There is no concatenation function. For example, these WHERE clauses would not work:

```
WHERE AFTER SALARY-AMT > BEFORE SALARY-AMT * 1.10
WHERE SUBSTR(AFTER FIRST-NAME 1,1) || '.' = "N."
```

## SQL Additions for Before and After Images

Standard SQL was created to extract data from a database, in which the data is always in its singular current state. With Tracer, you are dealing with BFOR and AFTR images of the same record. This adds a level of difficulty to the selection criteria. Consider this example:

```
SELECT ESALHIST-8002 WHERE SALARY-AMT > 2000
```

When running this request against a journal, does the user want all records where the new salary (after a MODIFY) is greater than 2000, or where the old salary (before a MODIFY) is greater than 2000?

Also, was the user intending to select newly stored records with salaries greater than 2000? What about salary records that were deleted where the salary was greater than 2000?

To avoid such confusion, these variable qualifiers are used by Tracer's SQL-like syntax to allow you to pull the exact data that you want.

Valid qualifiers are:

```
AFTER, BEFORE, EARLIEST, LATEST,
EITHER, BOTH, DELTA, DELTA%, IDELTA%
```

Default: If you omit a qualifier before a variable name, the default is *qualifier LATEST*

This is a detailed explanation of each qualifier:

- |       |   |
|-------|---|
| BFOR  | This is the value from the BEFORE image only (values of STOREs are treated as zero or blank). |
| AFTER | This is the value from the AFTER image only (values of ERASEs are treated as zero or blank).  |

Examples:

```
SELECT ESALHIST-8002 WHERE AFTER SALARY-AMT > 4000
```

Selects all stores and modifies where the new salary is greater than \$4000.

```
SELECT EMPLOYEE-8001 WHERE BEFORE STATE EQ 'OK'  
AND AFTER STATE EQ 'TX'
```

Selects all people whose state changed from OK to TX.

```
SELECT EMPLOYEE-8001 WHERE BEFORE STATE NE AFTER STATE
```

Selects all people whose state changed, including all stores and erases.

The next two qualifiers resolve the problem of treating ERASEs and STOREs as zero and blank. Thus, the LATEST and EARLIEST value are only different when the verb is a MODIFY.

**EARLIEST** For a MODIFY and ERASE, this is the value of the BEFORE image, but for STOREs it is the value of the AFTER image.

**LATEST** For a MODIFY and STORE, this is the value of the AFTER image, but for ERASEs it is the value of the BEFORE image.

This is the DEFAULT QUALIFIER if no qualifier is specified before an ELEMENT NAME.

Examples:

```
SELECT ESALHIST-8002 WHERE LATEST SALARY-AMT > 4000
```

Selects all stores and modifies where the new salary is greater than \$4000 and also all erases where the salary was greater then \$4000.

```
SELECT ESALHIST-8002 WHERE BEFORE SALARY-AMT  
NE AFTER SALARY-AMT
```

Selects all MODIFYs where the SALARY-AMT changed, and all STOREs and ERASEs where, for the STORE, the before images are blank or zero and, for the ERASE, the after images are blank or zero.

```
SELECT ESALHIST-8002 WHERE EARLIEST SALARY-AMT  
NE LATEST SALARY-AMT
```

This is a creative way of selecting only those updates made with the MODIFY verb where the SALARY changed. On STOREs and ERASEs, the EARLIEST and LATEST amounts would be identical, and therefore would not be selected.

These two qualifiers look at both the before and after images:

- EITHER - inserts both the BEFORE and AFTER image in the equation, resolves the equation twice, then, if the equation is true for either case, the equation is resolved as true. Otherwise, it is resolved as false. (In the case of STORE and ERASE, the result of the equation is treated as true for the missing image).
- BOTH - inserts both the BEFORE and AFTER image in the equation, resolves the equation twice, then, if the equation is true for both cases, the equation is resolved as true. Otherwise, it is resolved as false. (In the case of STOREs and ERASEs, the result of the equation is treated as true for the missing image).

Examples:

```
SELECT ESALHIST-8002 WHERE EITHER SALARY-AMT > 4000
```

Selects all journal records for the ESALHIST-8002 record where either the before SALARY-AMT was greater than \$4000 or the after SALARY-AMT is greater than \$4000.

```
SELECT ESALHIST-8002 WHERE AFTER SALARY-AMT > 4000  
AND BOTH JOB-TITLE-8002 = 'PROGRAMMER'
```

Selects all journal records for the ESALHIST-8002 record where the after SALARY-AMT was greater than \$4000 and the JOB-TITLE did not change and was equal to PROGRAMMER.

These DELTA qualifiers look at both the before and after images of numeric fields and compute the difference and/or the percentage difference:

- DELTA - the number derived by subtracting the before image from the after image (for example, before SALARY-AMT = 1000, after SALARY-AMT = 1200, the DELTA = +200).

DELTA is negative if the after image is less than the before image. On a STORE, DELTA is equal to the numeric value of the after image. On an ERASE, DELTA is equal to the negative value of the before image.

- DELTA% - the percentage increase (number created by dividing DELTA by the BEFORE image value). For example, before salary-amt = 1000, after SALARY-AMT = 1200, the DELTA = +200, and DELTA% = +20.

DELTA% is negative if the after image is less than the before image. Since you cannot divide by zero, STORE records are not selected. Internally, on a STORE, DELTA% equals 0 (see IDELTA% below to treat this as infinity). On an ERASE, DELTA% is equal to -100, since 100% of the entire amount is erased.

- IDELTA% - the number created by dividing DELTA (see above by the BEFORE image value). Division by zero is treated as infinity. Thus, when testing WHERE IDELTA% > any number, all STORE records that meet other selection criteria are selected. Internally, DELTA% is set to the number: 99,999,999.9999 for infinity.

Examples:

```
SELECT ESALHIST-8002 WHERE DELTA SALARY-AMT > 1000
OR DELTA% SALARY-AMT > 15
```

Selects all journal records for the ESALHIST-8002 record where the SALARY-AMT increased by either \$1000 or 15%.

```
SELECT ESALHIST-8002 WHERE DELTA SALARY-AMT <> 0
```

Tests whether the SALARY-AMT changed.

For ease of reference, any time you specify DELTA or DELTA% for a field-name, the DELTA/DELTA% values are printed on the Tracker report below the field-name/value.

This chart shows the results of these qualifiers with almost every possible combination. [Figure 21 on page 90](#) has 11 test columns that correspond to the values provided here.

Test	Test Description
1	IF BEFORE SALARY-AMT < 750
2	IF AFTER SALARY-AMT < 750
3	IF EARLIEST SALARY-AMT < 750
4	IF LATEST SALARY-AMT < 750
5	IF EITHER SALARY-AMT < 750
6	IF BOTH SALARY-AMT < 750
7	IF EITHER SALARY-AMT BETWEEN 550 AND 750
8	IF BOTH SALARY-AMT BETWEEN 550 AND 750
9	IF DELTA SALARY-AMT > 100
10	IF DELTA SALARY-AMT > 15
11	IF DELTA % SALARY-AMT > 15

**Figure 21 • Sample Tracker Report**

VERB		BEFORE SALARY	AFTER SALARY	DELTA%	TEST NUMBER											
					1	2	3	4	5	6	7	8	9	10	11	
MOD	1	800	900	12.5%	F	F	F	F	F	F	F	F	F	F	F	F
MOD	2	400	500	25%	T	T	T	T	T	T	T	F	F	T	T	
MOD	3	400	900	125%	T	F	T	F	T	F	F	F	T	T	T	
MOD	4	400	600	50%	T	T	T	T	T	T	T	F	T	T	T	
MOD	5	700	800	14%	T	F	T	F	T	F	F	F	F	F	T	
MOD	6	600	700	16.6%	T <sup>4</sup>	T <sup>4</sup>	T <sup>4</sup>	T <sup>4</sup>	T	T	T	T	F	T	T	
MOD	7	700	600	-16.6%	T <sup>4</sup>	T <sup>4</sup>	T <sup>4</sup>	T <sup>4</sup>	T	T	T	T	F	F <sup>5</sup>	F <sup>5</sup>	
STO	8	NA	400	?	T	T	T <sup>7</sup>	T <sup>7</sup>	T	T	F	F	T	F <sup>6</sup>	T <sup>6</sup>	
STO	9	NA	650	?	T	T	T <sup>7</sup>	T <sup>7</sup>	T	T	T	T	T	F <sup>6</sup>	T <sup>6</sup>	
STO	10	NA	900	?	T <sup>1</sup>	F	F <sup>7</sup>	F <sup>7</sup>	F	F	F	F	T	F <sup>6</sup>	T <sup>6</sup>	
ERA	11	400	NA	0	T	T <sup>2</sup>	T <sup>73</sup>	T <sup>73</sup>	T <sup>3</sup>	T <sup>3</sup>	F <sup>3</sup>	F <sup>3</sup>	F <sup>5</sup>	F	F	
ERA	12	650	NA	0	T	T <sup>2</sup>	T <sup>7</sup>	T <sup>7</sup>	T	T	T	T	F <sup>5</sup>	F	F	
ERA	13	900	NA	0	F	T <sup>2</sup>	F <sup>7</sup>	F <sup>7</sup>	F	F	F	F	F <sup>5</sup>	F	F	

These notes correspond to the superscript numbers on the report shown in [Figure 21](#).

- 1** This test number is true because the BEFORE image on a STORE is treated as a ZERO.
- 2** The same reasoning applies, except with an AFTER image on an ERASE (had it been IF AFTER SALARY-AMT > 900, all of these would have been False).
- 3** All tests for EARLIEST, LATEST, EITHER, and BOTH are the same for STORES and ERASEs.
- 4** For MODIFY verbs, EARLIEST is the same as BEFORE and LATEST is the same as AFTER.
- 5** The change here was more than the specified value, but since it was negative (i.e., less than -100), the result is False.
- 6** This shows the only difference between DELTA% and IDELTA%. IDELTA% treats division by zero as infinity, thus causing all STOREs to be selected.
- 7** EARLIEST and LATEST return the same value for STOREs and ERASEs.

## Selection Criteria

This section discusses the difference between selection criteria for the DBTRFMT (@TE0BREF) and DBTRRFMT (@TG0BRRF) utilities. Associated statistics reports are also shown for each.

## DBTRRFMT Parameters

The selection criteria for the DBTRRFMT program (the program that creates reports from the reformatted journal file) are listed below. The selection parameter names are the same as those for the DBTRFMT program. Since the parameter names are the same, you can generally run the same input parameters into either the DBTRFMT or the DBTRRFMT utilities.

Other than the ones listed below, SEL- parameters will be ignored by the DBTRRFMT utility. In other words, the DBTRFMT utility has many additional parameters that may be used to build the reformatted file. Many of these parameters are no longer applicable when running a report against the reformatted file.

These are the valid DBTRRFMT selection parameters:

```
SEL-RU-ENVIRONMENT
SEL-RU-BEGIN-DATE *
SEL-RU-END-DATE *
SEL-RU-BEGIN-TIME *
SEL-RU-END-TIME *
SEL-PROGRAM-MASK
SEL-VERBNAME-MASK
SEL-DDNAME-MASK
SEL-TASK-NAME-MASK
SEL-TERMINAL-MASK
SEL-USERID-MASK
SEL-SUBSCHEMA-MASK
SEL-TYPE-RUN-UNIT-SELECTION
SEL-RUN-UNIT-NUMBER-1
SEL-RUN-UNIT-NUMBER-2
SEL-TYPE-TASK-ID-SELECTION
SEL-TASK-ID-NUMBER-1
SEL-TASK-ID-NUMBER-2
SEL-SINGLE-RECORD-NAME
```

\* If you set the USE-ENDING-TIME-RECOVERY-UNIT parameter to Y in job @TF0BROL (see "[SEL-USE ENDING TIME RECOVERY UNIT](#)" on page 53), the dates used for this report are from the end check point of the recovery run unit.

See "[Selection Criteria Parameters](#)" on page 48 for complete descriptions of these parameters.

The DBTRRFMT utility displays a table of records rejected because of the previous selection types (see [Figure 27 on page 98](#)). For example, if the Tracer report is not produced and you are trying to determine why, you should look at this table. If all fields are zero except one, then you should check to see if you miskeyed or misspelled a selection parameter. For example, if you selected only program TRHH0010 and you accidentally typed TRHH0019, then you would probably not get a report.

### DBTRFMT Select and Reject Counters

These statistics displays can be found at the end of the //SYSOUT file when the DBTRFMT (@TE0BREF) utility is run. Though in actuality these statistics show as one continuous display, they have been broken into five sections so each can be discussed individually.

[Figure 22](#) shows the distribution of records returned from the DB/TRACE Assembler subroutine, based on the TR-O-RECORD-READY-FLAG. This might be used as a quick check to see what Tracer found on the journal. It enables you to quickly determine, for instance, how many BGIN (begin run unit), ENJ (end run unit), AREA (ready areas), COMT (commit) checkpoints were found on the journal, and how many total journal records were processed by Tracer.

**Figure 22 • Report: Distribution of Records**

```

*****
* THE FOLLOWING SHOWS THE DISTRIBUTION OF *
* RECORDS RETURNED FROM DB/TRACE ASSEM. SUBR.*
* BASED ON THE TR-O-RECORD-READY FLAG. *
*****
RECORDS READY                = 0000000016
*****
NO-RECORD-READY              = 0000012547
SR0 SELECTED                  = 0000000000
SR0 NOT SELECTED              = 0000000000
SELECTED CHECKPOINT BGIN      = 0000000298
SELECTED CHECKPOINT AREA      = 0000000921
SELECTED CHECKPOINT COMT      = 0000000029
SELECTED CHECKPOINT ABRT      = 0000000005
SELECTED CHECKPOINT ENDJ      = 0000000293
SELECTED CHECKPOINT UNKNOWN   = 0000000000
SKIPPED BGIN SYSTEM           = 0000000000
NOT SELECTED CHECKPOINT BGIN   = 0000000000
NOT SELECTED CHECKPOINT AREA   = 0000000000
NOT SELECTED CHECKPOINT COMT   = 0000000000
NOT SELECTED CHECKPOINT ABRT   = 0000000000
NOT SELECTED CHECKPOINT ENDJ   = 0000000000
NOT SELECTED CHECKPOINT UNKW   = 0000000000
USER STAMP RECORD              = 0000000000
USER SITE DBT-TO-DESPAN       = 0000000000
USER SITE DBT-NO-DESPAN       = 0000000000
PROBLEM RECORD                 = 0000000000
TASK INITIALIZATION            = 0000000000
TASK TERMINATION               = 0000000000
-----
*** TOTAL ***                 = 0000014109  TOTAL NOT READY = 0000001546
-----

```

[Figure 23](#) shows records that were not selected because of a SEL- variable on a specific value. You may have a parameter set, such as SEL-PROGRAM-MASK = ABC\*, which results in no records showing on your Tracer reports. You could scan this section for any large numbers, then verify that you really wanted a SEL- variable coded in the //DBTINPUT file for that value.

**Figure 23 • Rejected Records (Using SEL- Variable)**

```

*****
* THE FOLLOWING SHOWS THE NUMBER OF RECORDS *
* NOT SELECTED BECAUSE OF A SEL- VARIABLE *
* THAT SELECTS A SPECIFIC VALUE *
*****
SEL-RU-ENVIRONMENT           = 0000000000
SEL-YN-RETURN-SYSTEM-RECORDS = 0000000000
SEL-YN-RETURN-POINTERS      = 0000000000
SEL-YN-RETURN-ONLY-CHGD-4-MOD = 0000000000
SEL-PROGRAM-MASK           = 0000000000
SEL-VERBNAME-MASK          = 0000000000
SEL-DDNAME-MASK            = 0000000000
SEL-TASK-NAME-MASK         = 0000000000
SEL-TERMINAL-MASK          = 0000000000
SEL-USERID-MASK            = 0000000000
SEL-SUBSCHEMA-MASK         = 0000000000 OR SEL-EXEMPT-SUBSCHEMA-MASK
SEL-RU-BEGIN-DATE/END-DATE  = 0000000000
SEL-RU-BEGIN-TIME/END-TIME  = 0000000000
SEL-RUN-UNIT-NUMBER-1/-2    = 0000000000 AND SEL-TYPE-RUN-UNIT-SELECTION
SEL-TASK-ID-NUMBER-1/-2     = 0000000000 AND SEL-TYPE-TASK-ID-SELECTION
SEL-SINGLE-RECORDNAME        = 0000000000
*****
*** TOTAL SEL- REJECTIONS   = 0000000000
*****

```

Figure 24 shows advanced parameters, or SEL- parameters that could also cause Tracer to reject a large number of records. Usually, a simple change, such as switching the value of a -SEL variable from Y to N will cause these records to be selected. When Tracer rejects a record, the WHY-NOT code (see ["CA-IDMS System Exit 14 Advanced Customization Techniques." on page 189](#)) is set. This is simply a summary count by some of the most common WHY-NOT codes (some of which have been grouped together because they related to a SEL- variable).

Figure 24 • Advanced Parameters

```
*****
* ADVANCED PARAMETERS - THESE SEL- PARMS          *
* MAY ALSO ELIMINATE RECORDS FROM BEING          *
* SELECTED - AND CAN BE EASILY CHANGED BY        *
* ADJUSTING THE VALUE OF THESE PARAMETERS IN     *
* YOUR //DBTINPUT FILE.                          *
*****
SEL-DBKEY-PAGE-NUMBER          = 0000000000 SEE WHY-NOT-CODE 132-133
SEL-YN-ONLY-1-PAGE-GROUP      = 0000000000 SEE WHY-NOT-CODE 134
SEL-TYPE-TASK-ID-SELECTION    = 0000000000 SEE WHY-NOT-CODE 137
SEL-ELEM-VALUE-1-REC          = 0000000000 SEE WHY-NOT-CODE 139
SEL-YN-DBKEY-IN-ANY-PNTR-POS  = 0000000000 SEE WHY-NOT-CODE 140
SEL-YN-DBKEY-IN-CHGD-PTR-POS  = 0000000000 SEE WHY-NOT-CODE 141
SEL-NAME-OF-ONLY-CHANGED-SET  = 0000000000 SEE WHY-NOT-CODE 142-144
SEL-YN-ONLY-TARGET-OF-DML-RC  = 0000000651 SEE WHY-NOT-CODE 146-149
SEL-YN-PROCESS-RHDCRUAL-RUS   = 0000000000 SEE WHY-NOT-CODE 151
SEL-YN-PROCESS-USER-RUAL-TYP  = 0000000000 SEE WHY-NOT-CODE 152
SEL-YN-PROCESS-ENDEAVOR-RUS   = 0000000000 SEE WHY-NOT-CODE 153
-----
TOT ADVANCED SEL- REJECTIONS = 0000000651
-----
```

[Figure 25](#) shows other journal records that Tracer rejected. The SEL- parameters have no control over these rejected records. They are also based on the WHY-NOT codes.

**Figure 25 • Rejected Journal Records**

```

*****
OTHER RECORDS REJECTED FOR VARIOUS "TR-O-WHY-
NOT-SELECTED-CODE" <> 0 - SEE APPENDIX D OF
USER-GUIDE.  THE "DBTRACE" ASSEM. SUBR.
RETURNS THESE RECORDS TO THE COBOL PROGRAM
EVEN THOUGH THEY DID NOT GET WRITTEN TO THE
//DBTRFMTJ FILE.  "DBTRACE" IS CALLED AND
RETURNS A RECORD ONCE FOR EVERY RECORD ON
THE JOURNAL FILE.  SEE APPENDIX D FOR CODES.
*****
                                WHY-NOT
                                CODE
TIME STAMP                      = 0000000000 004
COMBINE BFOR/AFTR                = 0000000000 039
FRAGMENT TO HOME                 = 0000000000 007
1ST PART OF SPAN                 = 0000000000 022/023
1ST PART OF FRAG                 = 0000000000 029
JSEG                             = 0000000000 082 EVERY JOURNAL HAS ONE OF THESE
AREA READY                       = 0000000000 135
SR2                              = 0000000000 013
SR3                              = 0000000000 014
PART OF REJECTED BGIN            = 0000000000 150
NO BFOR FOR AFTR                 = 0000000000 041
NO BFOR FOR AFTR/DBKEY           = 0000000000 129
MISSING BFOR OR RHDCRUAL         = 0000000000 084
MERGING TASK STATS/IGNORE        = 0000000000 113
START-TIME BEFORE JRNL BGIN      = 0000000000 068-071
JOURNAL CKPT RECORD              = 0000000000 112
UNCATEGORIZED                    = 0000000000 MISC
                                -----
*** TOTAL WHY-NOT REJECTIONS = 0000000000
                                -----
*****
RECORDS NOT IN //DBTRECIN        = 0000010699 (BASED ON SCHEMAS SPECIFIED IN @STEP1)
*****
SEL-UPDATE-TIME/START/STOP      = 0000000004
                                NOTE: TIME RANGE IS ONLY SELECTION CRITERIA
                                NOT FILTERED OUT BY ASSEM. SUBROUTINE ABOVE.
                                THEREFORE THESE RECORDS ARE IN THE //DBTRFMTJ
                                FILE BUT NOT ON REPORTS.
*****
WE READ ALL JOURNAL RECS FROM ARCHIVE TAPE
USERRPT: DBTURPT9 TIMES-CALLED = 000000677 RUN-UNIT=0000461910 RECORD=DEPARTMENT
VERB=STORE      D/P=D
DBTRPT01 HAS UNDERSTOOD FINISH DIRECTIVE
DBTRPT01 FINISHED
DBTRPT02 FINISHED
DBTRPT11 FINISHED
DBTRPT05 FINISHED
*****
PROGRAM: O-DBTLOCKS *****
NBR RECORDS WRITTEN TO THE //DBTLOCKS FILE =0000000000
PROGRAM O-DBTLOCKS HAS JUST FINISHED
*****
DBTRKR: TIMES CALLED=00000013
5320-CALL-USERPROG:  PROG-NAME=USERTRKR CALL-YN=N

```

Figure 26 shows summary counts, or a summary of how many records Tracer found on the journal file. Also shown are categories of rejections, based on all of the statistics displayed above. This is the main and last section of the statistics report. Usually, you will want to look at it first (it is always at the very bottom of the //SYSOUT listing). Then, if you need additional information, you can back up and look at the prior sections.

Figure 26 • Summary Counts

```

*****
SUMMARY COUNTS OF RECORDS READ, PROCESSED AND PASSED VARIOUS SELECTION FILTERS

                                COUNT
    ARCHIVED JOURNAL RECORDS READ = 0000014109  BLOCKS READ = 0000000240
    RECORDS REJECTED BY //DBTRECIN FILE = -0000010699
    RECORDS REJECTED BY SEL- STATEMENTS = -0000000000  THESE ARE ITEMIZED IN A PREVIOUS TABLE
    RECS REJ. BY ADVANCED SEL- STATEMENTS = -0000000651  THESE ARE ITEMIZED IN A PREVIOUS TABLE
    RECORDS REJECTED - SEE WHY-NOT-CODE = -0000000000  THESE ARE ITEMIZED IN A PREVIOUS TABLE
    NON-REJECTED JRECS FROM ASSEM.SUB = 0000001562
    CHECK-POINT JRECS FROM ASSEM.SUB = -0000001546

                                DATA+PNTR    DATA    PNTR-ONLY    OTHER
    REFORMATTED JRECS FROM ASSEM.SUB = 0000000016 0000000008 0000000008 0000000000
    RECORDS REJECTED FOR UPDATE-TIME RANGE = -0000000004 (NUMBER OF TIMES TESTED=0000000016)
                                                NOTE: TIME RANGE IS ONLY SELECTION CRITERIA
                                                NOT FILTERED OUT BY ASSEM. SUBROUTINE ABOVE,
                                                THEREFORE THESE RECORDS ARE IN THE //DBTRFMTJ FILE
                                                BUT NOT ON REPORTS.
    RECORDS PASSING ALL NON-SQL SELECTION = 0000000012 0000000007 0000000005 0000000000

    TRACKER RECORDS PASSING //FLDINCL SEL = 0000000012 0000000007 0000000005 //FLDINCL WAS NOT IN USE
    TRACKER RECORDS PASSING //FLDSELS SEL = 0000000000 0000000000 0000000000 //FLDSELS WAS NOT IN USE

    RECORDS WRITTEN TO //DBTRFMTJ FILE = 0000000016 0000000008 0000000008

COUNTS OF CALLS TO VARIOUS SUBRPROGRAMS:

    CALLS TO    DBTRACE = 0000014110
    CALLS TO DB/TRACKER = 0000000012
    CALLS TO    REPORT01 = 0000001558
    CALLS TO    REPORT02 = 0000000122
    CALLS TO    REPORT05 = 0000000012
    CALLS TO    REPORT11 = 0000000012
    CALLS TO    DBTLOCKS = 0000000000
    CALLS TO USER-WRITTEN SELECT SUBPROG = 0000000000 NO USER-WRITTEN SELECTION SUBPROGRAM SPECIFIED
    CALLS TO USER-WRITTEN REPORT = 0000000676 USERRPT-SUBPROGRAM-NAME = DBTURPT9

*****
DBTRFMT HAS NOW FINISHED

```

The statistics indicate that Tracer looked at 14,109 records and selected 16 as relating to the employee subschema (see ["Step 2 - DMCL and Schema Mapping" on page 21](#)). Out of those 16 records, 4 were rejected because they were not in the specified start/stop time range, thus leaving 12 records to be written to the reports. Since the start/stop time does not affect the number of records written to the reformatted file, there were 16 records written there.

If the //FLDINCL or //FLDSEL options were in effect, they could have further narrowed down the number of qualifying records.

The optional //FLDINCL file allows you to further limit record types and fields displayed on the Tracker report. Then, the SQL statements from the //FLDSELS file are finally applied and only 4 records qualified for printing.

### ***DBTRRFMT Select and Reject Counters***

The output from the DBTRRFMT (@TG0BRRF) utility provides slightly different information. In this utility, the selection is done by a COBOL program, rather than by the Assembler DBTRACE subroutine. NUM REJECTED shows a count of how many records were rejected because of this SEL- variable (see [Figure 27 on page 98](#)). NUM TESTED shows how many times that parameter was tested (once a record has been rejected by one parameter, it will not be tested by any of the other parameters). The values you coded for your -SEL statements are provided as a quick reference.

Figure 27 • Rejected Records (Using SEL- Variable)

```

*****
* THE FOLLOWING SHOWS THE NUMBER OF RECORDS
* NOT SELECTED BECAUSE OF A SEL- VARIABLE
*****

              NUM      NUM      YOUR
              REJECTED  TESTED  VALUES
              -----  -----  -----
SEL-RU-ENVIRONMENT      = 00000000 00000000 ALL
SEL-PROGRAM-MASK 1 OR 2  = 00000000 00000000 ALLPROGS ALLPROGS
SEL-VERBNAME-MASK      = 00000000 00000000 ALLVERBS
SEL-DDNAME-MASK        = 00000000 00000000 ALLDDNMS
SEL-TASK-NAME-MASK     = 00000000 00000000 ALLTASKS
SEL-TERMINAL-MASK      = 00000000 00000000 ALLTERMS
SEL-USERID-MASK        = 00000000 00000000 ALLUSERS
SEL-SUBSCHEMA-MASK     = 00000000 00000016 ALLSSCNS OR EXEMPT:NOEXEMPT
SEL-RU-BEGIN-DATE/END-DATE = 00000000 00000016 ALLDATES TO MM/DD/YY
SEL-RU-BEGIN-TIME/END-TIME = 00000000 00000016 00000000000000 TO 00000000000000
SEL-RUN-UNIT-NUMBER-1/-2 = 00000000 00000000 00000000000000 00000000000000
SEL-TASK-ID-NUMBER-1/-2 = 00000000 00000000 00000000000000 00000000000000
SEL-SINGLE-RECORDNAME    = 00000000 00000000 ALL-SUBS-RECORDS

USER-WRITTEN SELECTION SUBPROG= 00000000 00000000 NOT USED

*****

SUMMARY COUNTS OF RECORDS READ, PROCESSED AND PASSED VARIOUS SELECTION FILTERS

              COMBINED  DATA  PNTR-ONLY
REFORMATED JOURNAL RECORDS READ = 00000016
RECORDS PASSING SEL- PARS = 00000016 00000008 00000008
RECORDS PASSING ALL NON-SQL SELECTION = 00000016 00000008 00000008 I.E. NUMBER OF RECS ON REPORT
11
              AND NUMBER OF RECS PASSED TO DB/TRACKER
              FOR SQL SELECTION
TRACKER RECORDS PASSING //FLDINCL SEL = 000000008 000000008 000000000 //FLDINCL WAS NOT IN USE
TRACKER RECORDS PASSING //FLDSELS SEL = 000000008 000000008 000000000 //FLDSELS WAS IN USE

              CALLS TO DB/TRACKER = 00000008
              CALLS TO REPORT11 = 00000016
CALLS TO USER-WRITTEN SELECT SUBPROG = 00000000 NO USER-WRITTEN SELECTION SUBPROGRAM SPECIFIED
CALLS TO USER-WRITTEN REPORT = 00000000 NO USER-WRITTEN REPORTING SUBPROGRAM SPECIFIED

*****
DBTRFMT HAS NOW FINISHED

```

**Note:** \_\_\_\_\_

The Assembler Tracer subroutine does the selection criteria in the DBTRFMT program, while the COBOL DBTSELCT program does the selection criteria in the DBTRRMFT program. Some of the parameters relate to types of records (rather than strictly selection parameters) to be output to the reformatted journal file. If any of these other parameters are input into the DBTRRFMT utility, they are ignored.

---

Fields not described in this publication should *not* be modified. The remaining fields are used in the Tracer system and any changes could have unpredictable results.



---

# 6

## Tracer Report Facility

---

The Tracer Report facility consists of the Tracker (Field-level) report, Reformatter (Record-level) reports, and Database (Tuning and Statistics) reports. This chapter explains the report facility in detail through these sections:

Section	Page
<a href="#">Tracker (Field-level) Report</a>	<a href="#">101</a>
<a href="#">Record-level (Reformatter) Reports</a>	<a href="#">108</a>
<a href="#">Database (Tuning and Statistics) Reports</a>	<a href="#">130</a>

---

### Tracker (Field-level) Report

The Tracker report shows the changes made to database records down to the individual record element (field) level. Each field has a separate line on which the field name and its associated before and after value is displayed.

The Record Comparison program uses the schema map created by DSIDDMAP to locate the record definition by matching the record ID found in the Tracer record to the record ID stored in the table.

A record header is printed whenever record data is determined to have changed. The record header includes the record name, the dbkey, the verb that caused the data change (STORE, MODIFY, or ERASE), and information regarding who changed the record and when it was changed.

Each field that is different can be displayed or printed by name with before and after conditions. Character fields containing more than 40 bytes are broken into separate lines for reporting. Numeric fields are edited only to suppress leading zeros (in all cases), and to properly insert the decimal point. Packed (COBOL computational-3) fields are verified for valid format and invalid fields are displayed in hexadecimal format. Zoned decimal fields are not distinguished from display (character) fields, and are not edited in any way.

Pages of a sample Tracker report are shown in [Figure 28](#) and [Figure 29 on page 103](#).

Figure 28 • Sample Tracker Report

```

RUN-DATE01/22/01 TIME 17:51                ALLEN SYSTEMS GROUP - TRACER                PAGE 1
                                           TRACKER - FIELD-LEVEL AUDIT REPORT
LVL  ----FIELD NAME -----SUB             -----BEFORE-----
-----AFTER-----

=====
*****
** DATE=01/22/01 TIME=15:45:31 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,022
***** RECORD=EMPLOYEE DBKEY=75012:0001 VERB=MODIFY RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL TERMNL=DEVDLW
*****
KEY EMP-ID-0415 SAME 0024 0024
02 EMP-ID-0415 SAME 0024 0024
02 EMP-NAME-0415 <GROUP> <GROUP>
03 EMP-FIRST-NAME-0415 SAME JANE JANE
03 EMP-LAST-NAME-0415 CHGD DOUGH NEWLASTNAME
02 EMP-ADDRESS-0415 <GROUP> <GROUP>
03 EMP-STREET-0415 SAME 32 PLACE ST 32PLACE ST
03 EMP-CITY-0415 SAME ANYTOWN ANYTOWN
03 EMP-STATE-0415 SAME OK OK
03 EMP-ZIP-0415 <GROUP> <GROUP>
04 EMP-ZIP-FIRST-FIVE-0415 SAME 12345 12345
04 EMP-ZIP-LAST-FOUR-0415 SAME 6789 6789
02 EMP-PHONE-0415 SAME 8009325536 8009325536
02 STATUS-0415 CHGD 01 02
02 SS-NUMBER-0415 SAME 022337878 022337878
02 START-DATE-0415 <GROUP> <GROUP>
03 START-YEAR-0415 SAME 76 76
03 START-MONTH-0415 SAME 08 08
03 START-DAY-0415 SAME 08 08
02 TERMINATION-DATE-0415 <GROUP> <GROUP>
03 TERMINATION-YEAR-0415 SAME 99 99
03 TERMINATION-MONTH-0415 SAME 12 12
03 TERMINATION-DAY-0415 SAME 31 31
02 BIRTH-DATE-0415 <GROUP> <GROUP>
03 BIRTH-YEAR-0415 SAME 51 51
03 BIRTH-MONTH-0415 SAME 03 03
03 BIRTH-DAY-0415 SAME 29 29
*****
** DATE=01/22/01 TIME=15:48:24 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,046
***** RECORD=EXPERTISE DBKEY=75012:0004 VERB=MODIFY RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL TERMNL=DEVDLW
***** VIA SET NAME= EMP-EXPERTISE
***** VIA SETS OWNER REC NAME = EMPLOYEE OWNER DBKEY= 75012:0001
*****
02 SKILL-LEVEL-0425 CHGD 04 05
02 EXPERTISE-DATE-0425 <GROUP> <GROUP>
03 EXPERTISE-YEAR-0425 SAME 72 72
03 EXPERTISE-MONTH-0425 SAME 01 01
03 EXPERTISE-DAY-0425 SAME 28 28
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,085
***** RECORD=EXPERTISE DBKEY=75012:0006 VERB=STORE RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL TERMNL=DEVDLW
***** VIA SET NAME= EMP-EXPERTISE
***** VIA SETS OWNER REC NAME = EMPLOYEE OWNER DBKEY= 75012:0001
*****
02 SKILL-LEVEL-0425 AFTR 07
02 EXPERTISE-DATE-0425 AFTR <GROUP> <GROUP>
03 EXPERTISE-YEAR-0425 AFTR 94
03 EXPERTISE-MONTH-0425 AFTR 06
    
```

Figure 29 • Sample Tracker Report - 132-Column Mode Version

```

RUN-DATE 01/22/01 TIME 17:51                ALLEN SYSTEMS GROUP - TRACER                PAGE 2
TRACKER - FIELD-LEVEL AUDIT REPORT
LVL  ----FIELD NAME ----- SUB -----BEFORE-----AFTER-----
=====
03 EXPERTISE-DAY-0425                AFTR                28
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,087
***** RECORD=SKILL                DBKEY=75168:0006                VERB=MODIFY RUN-UNIT=101,345                ENVIRON=ONLINE
***** USER=DEVDLW                SUBSCH=EMPSS01                TASK=DBOL                TERMNL=DEVDLW
*****
KEY SKILL-ID-0455                SAME 9170                9170
02 SKILL-ID-0455                SAME 9170                9170
02 SKILL-NAME-0455                SAME THINKING                THINKING
02 SKILL-DESCRIPTION-0455                CHGD                ABLE TO THINK BEFORE ACTING
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,091
***** RECORD=EMPLOYEE                DBKEY=75012:0001                VERB=DISCONCT RUN-UNIT=101,345                ENVIRON=ONLINE
***** USER=DEVDLW                SUBSCH=EMPSS01                TASK=DBOL                TERMNL=DEVDLW
***** CHANGED SET NAME= DEPT-EMPLOYEE NEXT DBKEY= 75160:0010
***** CHANGED SETS OWNER REC NAME = DEPARTMENT OWNER DBKEY= 75160:0010
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,093
***** RECORD=DEPARTMENT                DBKEY=75166:0001                VERB=MODIFY RUN-UNIT=101,345                ENVIRON=ONLINE
***** USER=DEVDLW                SUBSCH=EMPSS01                TASK=DBOL                TERMNL=DEVDLW
*****
KEY DEPT-ID-0410                SAME 3100                3100
02 DEPT-ID-0410                SAME 3100                3100
02 DEPT-NAME-0410                SAME GLOBAL SOFTWARE                GLOBAL SOFTWARE
02 DEPT-HEAD-ID-0410                CHGD 0003                0024
*****
** DATE=01/22/01 TIME=16:02:52 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,101
***** RECORD=EMPLOYEE                DBKEY=75012:0001                VERB=CONNECT RUN-UNIT=101,345                ENVIRON=ONLINE
***** USER=DEVDLW                SUBSCH=EMPSS01                TASK=DBOL                TERMNL=DEVDLW
***** CHANGED SET NAME= DEPT-EMPLOYEE NEXT DBKEY= 75007:0001
***** CHANGED SETS OWNER REC NAME = DEPARTMENT OWNER DBKEY= 75166:0001
*****
** DATE=01/22/01 TIME=16:04:09 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPDEMO J-SEQ=1,433,111
***** RECORD=JOB                DBKEY=75160:0011                VERB=STORE RUN-UNIT=101,345                ENVIRON=ONLINE
***** USER=DEVDLW                SUBSCH=EMPSS01                TASK=DBOL                TERMNL=DEVDLW
*****
KEY JOB-ID-0440                AFTR                7777
02 JOB-ID-0440                AFTR                7777
02 TITLE-0440                AFTR                V.P. GLOBAL SOFTWARE
02 DESCRIPTION-0440                AFTR <GROUP>                <GROUP>
03 DESCRIPTION-LINE-0440                (1) AFTR                DECIDE HOW TO MAKE SOFTWARE
03 DESCRIPTION-LINE-0440                (2) AFTR                FOR COMPANIES THROUGHOUT WORLD
02 REQUIREMENTS-0440                AFTR <GROUP>                <GROUP>
03 REQUIREMENT-LINE-0440                (1) AFTR                5-7 YEARS EXPERIENCE AS DBA, PROVING AB
                    AFTR+                ILITY TO THINK
03 REQUIREMENT-LINE-0440                (2) AFTR                AND ABILITY TO INTERFACE WITH PEOPL
                    AFTR+                E IN MANY DEPTS
02 MINIMUM-SALARY-0440                AFTR                550000.00
02 MAXIMUM-SALARY-0440                AFTR                700000.00
02 SALARY-GRADES-0440                (1) AFTR                01
02 SALARY-GRADES-0440                (2) AFTR                02
1
RUN-DATE 01/22/01 TIME 17:51                ALLEN SYSTEMS GROUP - TRACER

```

The report in [Figure 29](#) is the Tracker side-by-side, 132-column wide version. It is also possible to produce the Tracker report in 80-column mode, which stacks the after images under the before images so the report can be easily viewed on an 80-column screen or on narrow paper.

Figure 30 and Figure 31 on page 105 show the 80-column mode version of the Tracker report.

Figure 30 • Tracker Report - 80-Column Mode Version

```

RUN-DATE 01/22/01  TIME 17:48                ALLEN SYSTEMS GROUP - TRACER                PAGE 1
PAGE 1                                         TRACKER - FIELD-LEVEL AUDIT REPORT
-----FIELD NAME -----                     -----BEFORE/AFTER-----
*****
** DATE=01/22/01 TIME=15:45:31 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPEMO
***** RECORD=EMPLOYEE DBKEY=75012:0001 VERB=MODIFY
***** J-SEQ=1,433,022 RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL TERMNL=DEVDLW
*****
KEY EMP-ID-0415 SAME 0024
02 EMP-ID-0415 SAME 0024
02 EMP-NAME-0415 <GROUP>
03 EMP-FIRST-NAME-0415 SAME JANE
03 EMP-LAST-NAME-0415 BFOR DOUGH
02 EMP-ADDRESS-0415 AFTR NEWLASTNAME
03 EMP-STREET-0415 <GROUP>
03 EMP-CITY-0415 SAME 32 PLACE ST
03 EMP-STATE-0415 SAME ANYTOWN
03 EMP-ZIP-0415 <GROUP>
04 EMP-ZIP-FIRST-FIVE-0415 SAME 12345
04 EMP-ZIP-LAST-FOUR-0415 SAME 6789
02 EMP-PHONE-0415 SAME 8009325536
02 STATUS-0415 BFOR 01
AFTR 02
02 SS-NUMBER-0415 SAME 022337878
02 START-DATE-0415 <GROUP>
03 START-YEAR-0415 SAME 76
03 START-MONTH-0415 SAME 08
03 START-DAY-0415 SAME 08
02 TERMINATION-DATE-0415 <GROUP>
03 TERMINATION-YEAR-0415 SAME 99
03 TERMINATION-MONTH-0415 SAME 12
03 TERMINATION-DAY-0415 SAME 31
02 BIRTH-DATE-0415 <GROUP>
03 BIRTH-YEAR-0415 SAME 51
03 BIRTH-MONTH-0415 SAME 03
03 BIRTH-DAY-0415 SAME 29
*****
** DATE=01/22/01 TIME=15:48:24 PGM=DBOLMAIN PAGE-GRP= 0 SEGMENT=EMPEMO
***** RECORD=EXPERTISE DBKEY=75012:0004 VERB=MODIFY
***** J-SEQ=1,433,046 RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL TERMNL=DEVDLW
***** VIA SET NAME= EMP-EXPERTISE
***** VIA SETS OWNER REC NAME = EMPLOYEE OWNER DBKEY= 75012:0001
*****
02 SKILL-LEVEL-0425 BFOR 04
AFTR 05
02 EXPERTISE-DATE-0425 <GROUP>
03 EXPERTISE-YEAR-0425 SAME 72
03 EXPERTISE-MONTH-0425 SAME 01
03 EXPERTISE-DAY-0425 SAME 28

```

**Note:**

For convenience, the page number is printed on both the left and right side of the page.

Figure 31 • Tracer Report - 80 Column Mode Version

```

RUN-DATE 01/22/01 TIME 17:48                ALLEN SYSTEMS GROUP - TRACER                PAGE 2
PAGE 2                                        TRACKER - FIELD-LEVEL AUDIT REPORT
-----FIELD NAME -----BEFORE/AFTER-----
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE=GRP= 0 SEGMENT=EMPDEMO
***** RECORD=EXPERTISE DBKEY=75012:0006 VERB=STORE
***** J-SEQ=1,433,085 RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL TERMNL=DEVDLW
***** VIA SET NAME= EMP-EXPERTISE
***** VIA SETS OWNER REC NAME = EMPLOYEE OWNER DBKEY= 75012:0001
*****
02 SKILL-LEVEL-0425 AFTR
02 EXPERTISE-DATE-0425 AFTR <GROUP>
03 EXPERTISE-YEAR-0425 AFTR
03 EXPERTISE-MONTH-0425 AFTR
03 EXPERTISE-DAY-0425 AFTR
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE=GRP= 0 SEGMENT=EMPDEMO
***** RECORD=SKILL DBKEY=75168:0006 VERB=MODIFY
***** J-SEQ=1,433,087 RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL
TERMNL=DEVDLW
*****
KEY SKILL-ID-0455 SAME 9170
02 SKILL-ID-0455 SAME 9170
02 SKILL-NAME-0455 SAME THINKING
02 SKILL-DESCRIPTION-0455 BFOR
AFTR ABLE TO THINK BEFORE ACTING
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE=GRP= 0 SEGMENT=EMPDEMO
***** RECORD=EMPLOYEE DBKEY=75012:0001 VERB=DISCONCT
***** J-SEQ=1,433,091 RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL
TERMNL=DEVDLW
***** CHANGED SET NAME= DEPT-EMPLOYEE NEXT DBKEY= 75160:0010
***** CHANGED SETS OWNER REC NAME = DEPARTMENT OWNER DBKEY=
75160:0010
*****
** DATE=01/22/01 TIME=15:55:45 PGM=DBOLMAIN PAGE=GRP= 0 SEGMENT=EMPDEMO
***** RECORD=DEPARTMENT DBKEY=75166:0001 VERB=MODIFY
***** J-SEQ=1,433,093 RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL
TERMNL=DEVDLW
*****
KEY DEPT-ID-0410 SAME 3100
02 DEPT-ID-0410 SAME 3100
02 DEPT-NAME-0410 SAME GLOBAL SOFTWARE
02 DEPT-HEAD-ID-0410 BFOR 0003
AFTR 0024
*****
** DATE=01/22/01 TIME=16:02:52 PGM=DBOLMAIN PAGE=GRP= 0 SEGMENT=EMPDEMO
***** RECORD=EMPLOYEE DBKEY=75012:0001 VERB=CONNECT
***** J-SEQ=1,433,101 RUN-UNIT=101,345 ENVIRON=ONLINE
***** USER=DEVDLW SUBSCH=EMPSS01 TASK=DBOL TERMNL=DEVDLW
***** CHANGED SET NAME= DEPT-EMPLOYEE NEXT DBKEY= 75007:0001
***** CHANGED SETS OWNER REC NAME = DEPARTMENT OWNER DBKEY=
75166:0001
*****

```

This table describes the fields on the Tracer reports:

Field	Description
DATE	Date (month, day, and year) the update took place.
TIME	Time (in hours, minutes, and seconds) of update. HH/MM/SS format.
PGM	Program used to make the update.

Field	Description
PAGE-GRP	IDMS page group that was in effect when the updating program caused the update. IDMS uses the page group with the dbkey to uniquely identify the record that was updated.
SEGMENT	This field is present only when the journal tape was produced by a IDMS 12.x or 14.x system. The segment name is determined by Tracer at runtime by using information extracted from the global DMCL and matching the page group and page number of the updated record occurrence.
RECORD	Indicates which record type was modified.
DBKEY	Dbkey of the record occurrence that was modified.
VERB	IDMS verb used to modify the record (e.g., MODIFY, STORE, CONNECT, DISCONNECT, ERASE, ERASE ALL, ERASESEL, or ERASEPRM).
J-SEQ	Unique journal sequence number assigned by IDMS to this journal record. This number can be used to match updates in different detail reports.
RUN-UNIT	Unique number that IDMS assigns to the run unit that requested this update.
ENVIRON	Environment in which the updating program was executing (e.g., batch, online, CICS, etc.).
USER	User ID associated with the run unit that caused the update. You must have IDMS System Exit 14 installed for this field to be displayed.
SUBSCH	Subschema used to update this record. You must have IDMS System Exit 14 installed for this field to be displayed.
TASK	Task name associated with the online program. You must have IDMS System Exit 14 installed for this field to be displayed.
TERMNL	Physical terminal ID used to make this update.

---

Field	Description
	For updates to records with storage mode of VIA:
VIA SET NAME	If this record type is stored via some set, this is that via-set's name.
VIA SETS OWNER REC NAME	Name of the record defined as owner of the via set of the updated record.
	For updates that affect pointers only:
CHANGED SET NAME	Name of the set whose pointers changed due to an update.
NEXT DBKEY	The after image of a Next pointer that changed.
CHANGED SETS OWNER REC NAME	Name of the record that participates as owner of the set whose pointers changed.
OWNER DBKEY	If the changed set was defined in its schema with an owner pointer, this is the after image of the owner pointer for the set whose pointers were changed.
KEY	Identifies the field that serves as a key for that record type.
PT	Displays if pointers were updated. Before images of the pointer are prefixed by a B; after images of the pointer are prefixed by an A.
BFOR	The data that was in the field before the change was made.
AFTR	The data in the field after the change was made.
SAME	The field was not changed.

[Figure 32](#) shows an example of the Tracker report for CA-IDMS 12.x or 14.x SQL run units.

**Figure 32 • Sample Tracker Report - IDMS SQL Run Units**

TRACKER - FIELD-LEVEL AUDIT REPORT			
LVL	FIELD NAME	SUB	BEFORE-AFTER
***** SCHEMA=DEMOEMPL VERSION=0000 DDNAME=EMPLDEMO *****			
**	DATE=01/20/01	TIME=09:45:16	PGM=IDMSOCF PAGE-GRP= 0 SEGMENT=SQLDEMO J-SEQ=4,235,796
*****	RECORD=BENEFITS	DBKEY=77004:0004	VERB=MODIFY RUN-UNIT=160,366 ENVIRON=ONLINE
*****	USER=DEVDLW	TASK=OCF	TERMINL=DEVDLW
*****			
02	FISCAL_YEAR	SAME 2000	2000
02	EMP_ID	SAME 3411	3411
02	VAC_ACCRUED	SAME 68.00	68.00
02	VAC_TAKEN	SAME 68.00	68.00
02	SICK_ACCRUED	SAME 8.00	8.00
02	SICK_TAKEN	SAME 8.00	8.00
02	STOCK_PERCENT	SAME 0.000	0.000
02	STOCK_AMOUNT	SAME 0.00	0.00
02	LAST_REVIEW_DATE	SAME MON 1991-09-30 00:00:00.000000 273	MON 1991-09-30 00:00:00.000000 273
02	REVIEW_PERCENT	SAME 0.050	0.050
02	PROMO_DATE	SAME <NULL>	<NULL>
02	RETIRE_PLAN	CHGD BONDS	STOCK
02	RETIRE_PERCENT	SAME 0.030	0.030
02	BONUS_AMOUNT	SAME 1400.00	1400.00
02	COMP_ACCRUED	SAME 0.00	0.00
02	COMP_TAKEN	SAME 0.00	0.00
02	EDUC_LEVEL	SAME JRCOLL	JRCOLL
02	UNION_ID	SAME <NULL>	<NULL>
02	UNION_DUES	SAME <NULL>	<NULL>
*****			
***** SCHEMA=DEMOEMPL VERSION=0000 DDNAME=EMPLDEMO *****			
**	DATE=01/20/01	TIME=09:45:16	PGM=IDMSOCF PAGE-GRP= 0 SEGMENT=SQLDEMO J-SEQ=4,235,798
*****	RECORD=BENEFITS	DBKEY=77004:0005	VERB=MODIFY RUN-UNIT=160,366 ENVIRON=ONLINE
*****	USER=DEVDLW	TASK=OCF	TERMINL=DEVDLW
*****			
02	FISCAL_YEAR	SAME 1999	1999
02	EMP_ID	SAME 3411	3411
02	VAC_ACCRUED	SAME 120.00	120.00
02	VAC_TAKEN	SAME 120.00	120.00
02	SICK_ACCRUED	SAME 15.00	15.00
02	SICK_TAKEN	SAME 3.00	3.00
02	STOCK_PERCENT	SAME 0.000	0.000
02	STOCK_AMOUNT	SAME 0.00	0.00
02	LAST_REVIEW_DATE	SAME WED 1990-10-10 00:00:00.000000 283	WED 1990-10-10 00:00:00.000000 283
02	REVIEW_PERCENT	SAME 0.040	0.040
02	PROMO_DATE	SAME <NULL>	<NULL>
02	RETIRE_PLAN	CHGD BONDS	STOCK
02	RETIRE_PERCENT	SAME 0.030	0.030
02	BONUS_AMOUNT	SAME 1350.00	1350.00
02	COMP_ACCRUED	SAME 0.00	0.00
02	COMP_TAKEN	SAME 0.00	0.00

## Record-level (Reformatter) Reports

Tracer produces these record-level reports:

**Report 01.** Track Physical Record Updates

**Report 02.** DBOL/DMLO Updates for All Subschemas

**Report 05.** Provides information from CONNECT/DISCONNECT verbs

**Report 11.** Track Physical Record Updates (record level)

## Report 01 - Track Physical Record Updates

Report 01 tracks physical updates written to the IDMS journal archive file. This report provides detailed information about the physical updates IDMS performs for key online update transactions for the selected subschema.

Report 01 lists one line for each record (or matched pair of records) found on the reformatted journal file. It is produced by the DBTRFMT program (a program that reads the actual journal files and creates this and other reports, and, optionally, the reformatted journal file). Report 01 contains some journal record types not found on Report 11 (e.g., begin, commit, endjob).

SORT SEQUENCE: Date/Time (the order records are stored on the IDMS journal).

If you specify these selection criteria, Report 01 displays every dbkey (each physical page number) that IDMS updated for a specific run unit, including veiled updates to pointer positions and system records:

```
SEL-YN-RETURN-POINTERSY  
SEL-YN-RETURN-SYSTEM-RECORDSY  
SEL-TYPE-UPDTS-USER-SYS-ALLA  
SEL-YN-RETURN-ONLY-CHNGD-MODN
```

Complete selection criteria details are included in ["Control Parameters" on page 37](#).

Figure 33 and Figure 34 show a two-page sample of Tracer's Report 01.

Figure 33 • Sample Report 01 (page 1)

REPORT 01.										ALLEN SYSTEMS GROUP - TRACER										RUN DATE 1/22/01		PAGE	1
JOB:DEVDLW02/JOB01591										TRACER TRACK PHYSICAL RECORD UPDATES													
TP E	RUN	UNIT	TASK	ID	DEKEY-PG:	LINX	PROGRAM	D	VE	RECORD	PAGE	USERID	RFMT-R#	CHGD	STRT	U-TIME	BFOR	DATA	AFTR	DATA			
JR V	NUMBER	OR/PG-READ	OR/PGS-W	NAME	P	RB	NAME	GRUP	IF	X14	OFST	MMDD	HHMMSS	1234567890	1234567890								
BG	D	101345		31			DBOLMAIN			RUN UNIT BEGIN							0122	154146					
AF	D	101345		31	75160:0010		DBOLMAIN	P	MO	DEPARTMENT	0	DEVDLW	1	8	0122	154146	DEPT-EMPLOYEE		N=TG				
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	2	8	0122	154146	DEPT-EMPLOYEE		Y=TC				
AF	D	101345		31	75160:0010		DBOLMAIN	P	MO	DEPARTMENT	0	DEVDLW	3	8	0122	154146	DEPT-EMPLOYEE		N=TG				
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	4	8	0122	154146	DEPT-EMPLOYEE		Y=TG				
BO	D	101345		31	75002:0002		DBOLMAIN	D	MO	SR8-INDX	0	DEVDLW				1	0122	154147					
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	5	20	0122	154147	EMP-NAME-NDX		Y=TG				
BO	D	101345		31	75002:0004		DBOLMAIN	D	MO	SR8-INDX	0	DEVDLW				1	0122	154147					
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	6	20	0122	154147	EMP-NAME-NDX		Y=TG				
BO	D	101345		31	75158:0010		DBOLMAIN	D	MO	SR8-INDX	0	DEVDLW				1	0122	154147					
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	7	24	0122	154147	OFFICE-EMPLOYEE		Y=TG				
BO	D	101345		31	75158:0010		DBOLMAIN	D	MO	SR8-INDX	0	DEVDLW				1	0122	154147					
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	8	24	0122	154147	OFFICE-EMPLOYEE		Y=TG				
BO	D	101345		31	75012:0001		DBOLMAIN	D	MO	EMPLOYEE	0	DEVDLW	9	15	0122	154531	0024JANE	0024JANE					
AF	D	101345		31	75012:0003		DBOLMAIN	P	MO	EXPERTISE	0	DEVDLW	10	4	0122	154531	EMP-EXPERTISE		N=TG				
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	11	48	0122	154531	EMP-EXPERTISE		N=TG				
AF	D	101345		31	75012:0004		DBOLMAIN	P	MO	EXPERTISE	0	DEVDLW	12		0122	154531	EMP-EXPERTISE		Y=TG				
AF	D	101345		31	75012:0001		DBOLMAIN	P	MO	EMPLOYEE	0	DEVDLW	13	48	01220122	154531	EMP-EXPERTISE		N=TG				
AF	D	101345		31	75012:0003		DBOLMAIN	P	MO	EXPERTISE	0	DEVDLW	14	4	0122	154531	EMP-EXPERTISE		N=TG				
AF	D	101345		31	75012:0004		DBOLMAIN	P	MO	EXPERTISE	0	DEVDLW	15		0122	154531	EMP-EXPERTISE		Y=TG				
BO	D	101345		31	75163:0005		DBOLMAIN	D	MO	SR8-INDX	0	DEVDLW				1	0122	154531					
AF	D	101345		31	75012:0004		DBOLMAIN	P	MO	EXPERTISE	0	DEVDLW	16	12	0122	154531	SKILL-EXPERTISE		Y=TG				
BO	D	101345		31	75163:0005		DBOLMAIN	D	MO	SR8-INDX	0	DEVDLW				1	0122	154824					
AF	D	101345		31	75012:0004		DBOLMAIN	P	MO	EXPERTISE	0	DEVDLW	17	12	0122	154824	SKILL-EXPERTISE		Y=TG				
BO	D	101345		31	75012:0004		DBOLMAIN	D	MO	EXPERTISE	0	DEVDLW	18	2	0122	154824	04720128	05720128					
AF	D	101345		31	75012:0001		DBOLMAIN	P	ST	EMPLOYEE	0	DEVDLW	19	48	0122	155545	EMP-EXPERTISE		N=TG				
AF	D	101345		31	75012:0004		DBOLMAIN	P	ST	EXPERTISE	0	DEVDLW	20	4	0122	155545	EMP-EXPERTISE		N=TG				
BO	D	101345		31	75163:0005		DBOLMAIN	D	ST	SR8-INDX	0	DEVDLW				20	0122	155545					
AF	D	101345		31	75012:0006		DBOLMAIN	D	ST	EXPERTISE	0	DEVDLW	21		0122	155545		07940122					
BO	D	101345		31	75168:0006		DBOLMAIN	D	MO	SKILL	0	DEVDLW	22	17	0122	155545	9170THINKI	9170THINKI					
SM	D	101345		31	75160:0010		DBOLMAIN	P	DI	DEPARTMENT	0	DEVDLW	23	8	0122	155545	DEPT-EMPLOYEE		N=TG				
SM	D	101345		31	75012:0001		DBOLMAIN	P	DI	EMPLOYEE	0	DEVDLW	24	8	0122	155545	DEPT-EMPLOYEE		Y=TG				
BO	D	101345		31	75166:0001		DBOLMAIN	D	MO	DEPARTMENT	0	DEVDLW	25	52	0122	155545	3100GLOBAL	3100GLOBAL					
SM	D	101345		31	75040:0001		DBOLMAIN	P	CO	EMPLOYEE	0	DEVDLW	26	8	0122	160252	DEPT-EMPLOYEE		N=TG				
SM	D	101345		31	75007:0001		DBOLMAIN	P	CO	EMPLOYEE	0	DEVDLW	27	12	0122	160252	DEPT-EMPLOYEE		N=TG				
SM	D	101345		31	75012:0001		DBOLMAIN	P	CO	EMPLOYEE	0	DEVDLW	28	8	0122	160252	DEPT-EMPLOYEE		Y=TG				
AF	D	101345		31	75160:0010		DBOLMAIN	P	ST	DEPARTMENT	0	DEVDLW	29		0122	160252	CALC		N=TG				
AF	D	101345		31	75160:0000		DBOLMAIN	P	ST	SR1-CALC	0	DEVDLW			4	0122	160252	SR1-REC-CALC-SET		N=TG			
BO	D	101345		31	75154:0004		DBOLMAIN	D	ST	SR8-INDX	0	DEVDLW				20	0122	160409					
AF	D	101345		31	75160:0011		DBOLMAIN	D	ST	JOB	0	DEVDLW	30		0122	160409		7777V.P.G					
AF	D	101345		31	75012:0001		DBOLMAIN	P	ST	EMPLOYEE	0	DEVDLW	31	40	01220122	161929	EMP-EMPOSITION		N=TG				
AF	D	101345		31	75012:0002		DBOLMAIN	P	ST	EMPOSITION	0	DEVDLW	32	4	0122	161929	EMP-EMPOSITION		N=TG				
AF	D	101345		31	75012:0007		DBOLMAIN	D	ST	EMPOSITION	0	DEVDLW	33		0122	161929		9401229912					
SM	D	101345		31	75160:0011		DBOLMAIN	P	CO	JOB	0	DEVDLW	34	12	0122	161929	JOB-EMPOSITION						

Figure 34 • Sample Report 01 (page 2)

REPORT 01.										ALLEN SYSTEMS GROUP - TRACER										RUN DATE 1/22/01		PAGE	2
JOB:DEVDLW02/JOB01591										TRACER TRACK PHYSICAL RECORD UPDATES													
TP E	RUN	UNIT	TASK	ID	DEKEY-PG:	LINX	PROGRAM	D	VE	RECORD	PAGE	USERID	RFMT-R#	CHGD	STRT	U-TIME	BFOR	DATA	AFTR	DATA			
JR V	NUMBER	OR/PG-READ	OR/PGS-W	NAME	P	RB	NAME	GRUP	IF	X14	OFST	MMDD	HHMMSS	1234567890	1234567890								
AF	D	101345		31	75012:0001		DBOLMAIN	P	ST	EMPLOYEE	0	DEVDLW	45	56	0122	162354	MANAGES		N=TG				
AF	D	101345		31	75012:0008		DBOLMAIN	D	ST	STRUCTURE	0	DEVDLW	46		0122	162354		A	940122				
SM	D	101345		31	75040:0006		DBOLMAIN	P	CO	EMPLOYEE	0	DEVDLW	47	64	0122	162354	REPORTS-TO		N=TG				
SM	D	101345		31	75012:0008		DBOLMAIN	P	CO	STRUCTURE	0	DEVDLW	48	12	01220122	162354	REPORTS-TO		Y=TG				
EJ	D	101345		33	15		DBOLMAIN			NORMAL END		DEVDLW				0122	162354						

This section describes each field in Report 01.

**TP****JR**

Represents the type of IDMS journal record from which the displayed information was derived. These are the possible values:

- B4 = BFOR journal image
- AF = AFTR journal image
- BO = Both BFOR and AFTR journal images
- CM = COMT (Commit) journal image
- EJ = ENDJ (Normal end of run unit) journal image
- AB = ABRT (Aborted run unit) journal image
- BG = Begin record journal image
- SM = SAME (Connect or Disconnect journal image)

**E****V**

Represents the environment code. Identifies whether the run unit was initiated by a program which was loaded into a IDMS program pool or another region in the operating system (online or batch). This one-byte code is the first digit of the first four bytes of the local ID internal variable subschema field. These are the possible values:

- D = DBDC
- B = BATC (Batch)
- C = CICS

**RUN UNIT****NUMBER**

Displays the unique number that IDMS assigns to each run unit.

**TASK ID****OR/PG-READ**

Displays the unique number that IDMS assigns each TASK when the updating run unit is from an internal online environment. This is 0 if the run unit is from an external environment (i.e., something other than the IDMS-CV address space). Alternatively, when the record type is ENDJ, ABRT, or COMT, this column contains the number of pages read by IDMS.

**DBKEY-PG  
OR/PGS-W**

Displays the PAGE NUMBER of the dbkey for the record that was updated. This is the correct page number, even if you are using a non-standard matrix point in the schema that was associated with the updating subschema. Alternatively, when the record type is ENDJ, ABRT, or COMT, this column contains the number of pages written by IDMS.

**LINX**

Denotes the line index number of the dbkey for the record that was updated. This is the correct line index number, even if you are using a non-standard matrix point in the schema that was associated with the updating subschema.

**PROGRAM  
NAME**

Indicates the name that was in the PROGRAM-NAME field of the SUBSCHEMA-CTRL communication area when the program issued the BIND RUN UNIT verb.

**D  
P**

Represents a code that indicates whether the data portion, or only the pointer portion in this record was changed. These are the valid values:

- P = Only pointer changes were affected on this record.
- D = Indicates that the data portion of this record was journaled due to an update.

**VE  
RB**

Indicates the kind of update verb that caused this update to take place. These are the valid values

- ST = STORE
- ER = ERASE
- ES = ERASE SELECTIVE
- EP = ERASE PERMANENT
- EA = ERASE All
- MO = MODIFY
- CO = CONNECT
- DI = DISCONNECT
- AU = Auto Update (IDMS initiated)

When an application program reads a database area in one of the update modes, IDMS will sometimes execute some update cleanup jobs even though the program does not specifically tell IDMS to update anything. Examples include bringing fragments of record occurrences back to the home page, where space is now available.

**RECORD  
NAME**

Shows the 16-character name of the record that was modified. Tracer retrieves this name from the //DBTRECIN file that was built during setup.

**PAGE  
GRUP**

Displays the page group number in order to make a dbkey unique in shops that have segmented databases or re-use page ranges when defining their database areas.

**USERID  
IF X14**

Displays the first eight characters of the user ID associated with the run unit that caused the update. You must have installed the CA-IDMS System Exit 14 (RHDCUX14) in order for this field to be set.

**RFMT-R#**

Indicates the sequence number on the DBTRFMT file. This is the record number on which this record can be found in the DBTRFMT file. (It can be used when using TSO to browse that file.)

**CHGD  
OFST**

Displays the position in the record occurrence of the first data character that was changed. For journal records where only the pointers are journaled, this is the offset into the pointers (set dbkeys) of the first changed byte.

**STRT  
MMDD**

Displays the month and day on the BGIN record for the run unit that caused the update.

**U-TIME  
HHMMSS**

Displays the closest time (in hours, minutes, and seconds) that can be determined for the actual update occurrence. Tracer uses the closest prior TIME journal records to obtain the most accurate value possible.

**JRNL SEQ  
NUMBER**

Shows the unique journal sequence number assigned to SEQ NUMBER for this journal record. Can be used to match an update from this summary report to a Tracker report.

**BFOR DATA**  
**1234567890**

Displays the first 10 to 20 bytes of the data portion of the modified record when this line was built from a BFOR journal record. (Only applicable when the verb is MODIFY or DELETE.) To display 20 bytes, you must specify this input parameter:

RECRPT-COLUMNS            165

**AFTR DATA**  
**1234567890**

Displays the first 10-20 bytes of the data portion of the modified record when this line was built from an AFTR journal record. (Only applicable when the verb is MODIFY or STORE.) To display 20 bytes, you must specify this input parameter:

RECRPT-COLUMNS            165

**Report 02 - DBOL and DMLO Updates Any Subschema**

Report 02 lists one line for each update that was performed via any subschema through DBOL or DMLO.

Report 02 (DBTRPT02) tracks all updates made by DBOL or DMLO. It provides an account of all records on the IDMS journal archive file in which the BIND RUN UNIT command was issued by DBOL or DMLO. DBOL is a software package marketed and supported by Allen Systems Group, Inc. It provides online access and updating of the IDMS database. Notice that all records are printed regardless of whether they apply to the selection criteria in DBTRFMT.

SORT SEQUENCE: Date/Time (the order that records are stored on the IDMS journal).

[Figure 35 on page 115](#) shows a two-page sample of Tracer's Report 02. You invoke Report 02 by specifying Y for PRINT REPORT 02 in the DBTINPUT DD for the DBTRFMT program.

Figure 35 • Sample Report 02 (page 1)

REPORT 02.		ALLEN SYSTEMS GROUP - TRACER										RUN DATE	1/22/01	PAGE	1		
TP	RUN UNIT	DBKEY-PAGE:LNK	P	WHY	UPDATE	RCRD	STRT	U-TIME	USERID	SUBSCHEM	PGRP	R	1ST-15-BYTES	S-JOURNAL	TERMINAL	PROGRAM	
JR	NUMBER	OR/PG-WRT	R	NOT	VERBNAME	ID-#	MMDD	HMMSS	IF-UX14	IF-UX14	O-PR	S	123456789012345	SEQUENCE#			
BG	101345		B		--BEGIN-	0122	154146	DEVDLW	EMPSS01					1432985	DEVDLW	DBOLMAIN	
AR	101345	INS-DEMO-REGION	SHR	UPD		75101<LOPG	154146	DEVDLW	EMPSS01				75125<HIPG	1432987	DEVDLW	DBOLMAIN	
AR	101345	ORG-DEMO-REGION	SHR	UPD		75151<LOPG	154146	DEVDLW	EMPSS01				75175<HIPG	1432988	DEVDLW	DBOLMAIN	
AF	101345	75002:0002	N	22	MODIFY	8	0122	154146	DEVDLW	EMPSS01				1432998	DEVDLW	DBOLMAIN	
AF	101345	75002:0004	N	22	MODIFY	8	0122	154147	DEVDLW	EMPSS01				1433004	DEVDLW	DBOLMAIN	
AF	101345	75158:0010	Y		MODIFY	8	0122	154147	DEVDLW	EMPSS01				1433010	DEVDLW	DBOLMAIN	
AF	101345	75158:0010	Y		MODIFY	8	0122	154147	DEVDLW	EMPSS01				1433016	DEVDLW	DBOLMAIN	
AF	101345	75012:0001	Y		MODIFY	415	0122	154531	DEVDLW	EMPSS01		S	0024JANE	N	1433022	DEVDLW	DBOLMAIN
AF	101345	75163:0005	Y		MODIFY	8	0122	154531	DEVDLW	EMPSS01				1433036	DEVDLW	DBOLMAIN	
AF	101345	75163:0005	N	22	MODIFY	8	0122	154531	DEVDLW	EMPSS01				1433040	DEVDLW	DBOLMAIN	
AF	101345	75012:0004	Y		MODIFY	425	0122	154824	DEVDLW	EMPSS01			05720128	1433046	DEVDLW	DBOLMAIN	
AF	101345	75163:0005	Y		STORE	8	0122	155545	DEVDLW	EMPSS01				1433084	DEVDLW	DBOLMAIN	
AF	101345	75012:0006	Y		STORE	425	0122	155545	DEVDLW	EMPSS01			07940122	1433085	DEVDLW	DBOLMAIN	
AF	101345	75168:0006	Y		MODIFY	455	0122	155545	DEVDLW	EMPSS01			9170THINKING	1433087	DEVDLW	DBOLMAIN	
AF	101345	75160:0010	Y		DISCONCT	410	0122	155545	DEVDLW	EMPSS01				1433089	DEVDLW	DBOLMAIN	
AF	101345	75012:0001	Y		DISCONCT	415	0122	155545	DEVDLW	EMPSS01				1433091	DEVDLW	DBOLMAIN	
AF	101345	75166:0001	Y		MODIFY	410	0122	155545	DEVDLW	EMPSS01			3100GLOBAL SOFT	1433093	DEVDLW	DBOLMAIN	
AF	101345	75040:0001	N	22	CONNECT	415	0122	155545	DEVDLW	EMPSS01				1433095	DEVDLW	DBOLMAIN	
AF	101345	75040:0001	Y		CONNECT	415	0122	160252	DEVDLW	EMPSS01				1433097	DEVDLW	DBOLMAIN	
AF	101345	75007:0001	Y		CONNECT	415	0122	160252	DEVDLW	EMPSS01				1433099	DEVDLW	DBOLMAIN	
AF	101345	75012:0001	Y		CONNECT	415	0122	160252	DEVDLW	EMPSS01				1433101	DEVDLW	DBOLMAIN	
AF	101345	75154:0004	N	22	STORE	8	0122	160252	DEVDLW	EMPSS01				1433108	DEVDLW	DBOLMAIN	
AF	101345	75160:0011	Y		STORE	440	0122	160409	DEVDLW	EMPSS01			7777V.P. GL	1433111	DEVDLW	DBOLMAIN	
AF	101345	75012:0007	Y		STORE	420	0122	161929	DEVDLW	EMPSS01			94012299123142	1433148	DEVDLW	DBOLMAIN	
AF	101345	75160:0011	Y		CONNECT	440	0122	161929	DEVDLW	EMPSS01				1433150	DEVDLW	DBOLMAIN	
AF	101345	75012:0007	Y		CONNECT	420	0122	161929	DEVDLW	EMPSS01				1433152	DEVDLW	DBOLMAIN	
AF	101345	75022:0001	Y		MODIFY	415	0122	161929	DEVDLW	EMPSS01			0003JENNIFER G	1433154	DEVDLW	DBOLMAIN	
B4	101345	75022:0016	Y		ERASEALL	460	0122	161929	DEVDLW	EMPSS01			A 820101	1433155	DEVDLW	DBOLMAIN	
AF	101345	75002:0004	N	22	STORE	8	0122	161929	DEVDLW	EMPSS01				1433174	DEVDLW	DBOLMAIN	
AF	101345	75174:0012	N	22	STORE	8	0122	161929	DEVDLW	EMPSS01				1433178	DEVDLW	DBOLMAIN	
AF	101345	75040:0006	Y		STORE	415	0122	161954	DEVDLW	EMPSS01			7777SELL	S	1433181	DEVDLW	DBOLMAIN
AF	101345	75012:0008	Y		STORE	460	0122	162354	DEVDLW	EMPSS01			A 940122	1433216	DEVDLW	DBOLMAIN	
AF	101345	75040:0006	Y		CONNECT	415	0122	162354	DEVDLW	EMPSS01				1433218	DEVDLW	DBOLMAIN	
AF	101345	75012:0008	Y		CONNECT	460	0122	162354	DEVDLW	EMPSS01				1433220	DEVDLW	DBOLMAIN	
EJ	101345	15	E		--ENDJOB-	0122	162354	DEVDLW	EMPSS01					1433222	DEVDLW	DBOLMAIN	

This section describes each field in Report 02.

## TP JR

Represents the type of IDMS journal record from which the information was derived.  
Possible values are:

- AB = ABRT (aborted run unit) journal image
- AF = AFTR journal image
- BG = BGIN journal image
- B4 = BFOR journal image
- CM = COMT journal image
- EJ = ENDJ journal image

## RUN UNIT NUMBER

Displays the unique number that IDMS assigns to each run unit.

**DBKEY-PAGE:LN  
X  
OR/PG-WRT**

Displays the page number of the dbkey for the record that was updated. This is the correct page number, even if you are using a non-standard matrix point in the schema which was associated with the updating subschema. Alternatively, when the record type is ENDJ, ABRT, or COMT, this column contains the number of pages written by IDMS.

**P  
R**

Denotes whether or not this journal record was written to the reformatted file (DBTRFMTJ). Possible values are Y (Yes) or N (No).

**WHY  
NOT**

If the P/R columns contains an N, indicating that the record was not written to the reformatted file, this column contains a WHY NOT numeric code to help you determine the reason for its omission. See ["CA-IDMS System Exit 14 Advanced Customization Techniques" on page 189](#) for a full description of WHY NOT codes.

**UPDATE  
VERBNAME**

Displays, in standard IDMS names, the verb commands that caused this update. These are the available values:

ABORT  
AUTOUPDT  
BEGIN  
COMMIT  
CONNECT  
DISCONNECT  
END JOB  
ERASE  
ERASEALL  
ERASEPRM  
ERASESEL  
MODIFY  
STORE

**RCRD  
ID#**

Signifies the four-byte record ID number that was defined in the schema for the record type that was updated.

**STRT  
MMDD**

Displays the month and day on the Begin (BGIN) run unit record for the run unit that caused the update.

**U-TIME**  
**HHMMSS**

Displays the closest time that can be determined for the actual update occurrence. Tracer uses the closest prior TIME journal records to obtain the most accurate value possible. On an active system, it is at least within one or two seconds of the actual update time.

**USERID**  
**IF-UX14**

Displays the first eight characters of the user ID associated with the run unit that caused the update. You must have the CA-IDMS System Exit 14 (RHDCUX14) installed for this field to be set.

**SUBSCHEM**  
**IF-UX14**

Displays the subschema used to update this record. You must have the CA-IDMS System Exit 14 (RHDCUX14) installed for this field to be set.

**PGRP**  
**O-PR**

Displays the page group number in order to make a dbkey unique in shops that have segmented databases or re-use page ranges when defining their database areas.

**R**  
**S**

Signifies the second part of a spanned record if an S is displayed in this column. Information about when the first part of a spanned record is received is not currently available.

**1ST-15-BYTES**  
**0123456789012345**

Displays the first 15 bytes of the record's data. For spanned records, the first 15 bytes are also shown, but these bytes may be spanned as well.

**S-JOURNAL**  
**SEQUENCE#**

Displays the unique journal sequence number assigned by IDMS to this journal record. It can be used to match an update from this summary report to a Tracker report.

**TERMINAL**

Physical terminal ID that was used to make this update.

**PROGRAM**

Indicates the name which was in the PROGRAM-NAME field of the SUBSCHEMA-CTRL communication area when the program issued the BIND RUN UNIT verb.

## Report 05 - Track Set Updates

The main purpose of this report is to provide thorough information about updates done by either a CONNECT or a DISCONNECT verb and the pointers that are changed in set relationships (these are changed automatically by IDMS).

When set relationships change, IDMS updates the pointer portion of the database record (prefix) but not the data portion. In these cases, only the before and after image of the pointers (set dbkeys) for the record are written to the journal.

Thus, to obtain the full benefit of Report 05, you should set these parameters in the DBTINPUT ddname parameter file as indicated:

```
SEL-YN-RETURN-POINTERSY  
SEL-YN-RETURN-SYSTEM-RECORDSY  
SEL-YN-RETURN-ONLY-CHNGD-MODY  
SEL-YN-COMBINE-BFOR-AFTR-OUTY  
SEL-TYPE-UPDTS-USER-SYS-ALLA  
SEL-YN-ONLY-TARGET-OF-DML-RCN  
SEL-YN-TARGET-PTRO-NOT-CON-DIY
```

**Note:** \_\_\_\_\_

Only some of the columns in this report are filled in when the updated record has some of the data portion journaled, including the first 18 bytes of the data (in straight display) and the first nine bytes displayed in hexadecimal format.

---

When a CONNECT or DISCONNECT verb is issued, pointers must be updated for the record that is being connected and for any other records in the set that are logically positioned adjacent to that record.

IDMS automatically updates the appropriate record occurrences based on set relationship information. Such information is contained in the subschema load module specified by the updating program. Tracer interrogates the DBTRECIN file and extracts information about the set associated with the pointers that were changed.

Set-related information is provided on this new report and also written to the Tracer produced reformatted journal archive file (ddname is DBTRFMTJ) so it may be accessed by user-written programs.

The changed set information may also be utilized by user-written programs when you use the reformatted file as a transaction file to do incremental updates to a parallel file. This update keeps the file synchronized with the master IDMS database.

Two types of lines may be printed: Begin type (B) or Detail type (D). The Begin type line displays information about the updating run unit, including the program name, date, time, user ID, updating subschema, task name and number, terminal, and environment (batch (BATC), online (DBDC), or CICS (CICS)). The Detail type line displays information about the specific record occurrence that was on the IDMS journal archive tape. The fields on the Detail type line are arranged under the column headings at the top of each page.

[Figure 36](#) and [Figure 37 on page 120](#) show a two-page sample of Tracer's Report 05.

Figure 36 • Sample Report 05 (page 1)

REPORT 05.	ALLEN SYSTEMS GROUP - TRACER	TRACER TRACK SET AND DATA RECORD UPDATES	RUN DATE 1/22/01	PAGE 1								
T UPDATING	UPDATED-RECORDS DGT VE	UPDATED-RECORD	AU R TY ST S	CHANGED-SET-NAME	SET-POINTER.WW	SET-POINTER.WW	SEQUENCE-#	S				
L RUN-UNIT#	DBKEY-PAGE-LINE	IPG RB	VB S ST DC O	OR-DATA-IN-CHAR.	IP	OR-DATA-IN-HEX	.IP	FROM-SRC-J P				
B	101345	PRG>DBOLMAIN DT>01/22/01	TM>154146 PG>	0	USR>DEVLDLW	SUB>EMPSS01	ENV>DBDC	TSK>DBOL	TRM>DEVLDLW	TS#>	31	
D	101345	75160:0010	1P- MO DEPARTMENT	AJ	O CH	S DEPT-EMPLOYEE	75160:0010.AN	75012:0001.BN			1432990	P
D	101345	75012:0001	1P+ MO EMPLOYEE	DI	M CH	OA S DEPT-EMPLOYEE	75160:0010.BO	75160:0010.BN			1432992	A
D	101345	75160:0010	1P- MO DEPARTMENT	AJ	O CH	S DEPT-EMPLOYEE	75012:0001.AN	75160:0010.BN			1432994	P
D	101345	75012:0001	1P+ MO EMPLOYEE	CO	M CH	OA S DEPT-EMPLOYEE	75160:0010.AO	75160:0010.AN			1432996	A
D	101345	75002:0002	1D- MO SR8-INDX	AJ	A I	SY A SR8-INDEX-SET	75002:0001.AO	75002:0003.AP			1433000	Z
D	101345	75012:0001	1P+ MO EMPLOYEE	DI	M IS	OA S EMP-NAME-NDX	75002:0001.BO	75002:0002.BN			1433002	N
D	101345	75002:0004	1D- MO SR8-INDX	AJ	A I	SY A SR8-INDEX-SET	75002:0001.AO	75002:0002.AP			1433006	Z
D	101345	75012:0001	1P+ MO EMPLOYEE	CO	M IS	OA S EMP-NAME-NDX	75002:0001.AO	75002:0004.AN			1433008	N
D	101345	75158:0010	1D- MO SR8-INDX	AJ	A I	SY A SR8-INDEX-SET	75158:0002.AO	75158:0002.AP			1433010	Z
D	101345	75012:0001	1P+ MO EMPLOYEE	DI	M IU	OA S OFFICE-EMPLOYEE	75158:0002.BO	75158:0010.BN			1433014	Q
D	101345	75158:0010	1D- MO SR8-INDX	AJ	A I	SY A SR8-INDEX-SET	75158:0002.AO	75158:0002.AP			1433016	Z
D	101345	75012:0001	1P+ MO EMPLOYEE	CO	M IU	OA S OFFICE-EMPLOYEE	75158:0002.AO	75158:0010.AN			1433018	Q
D	101345	75012:0001	1D+ MO EMPLOYEE				0024JANE	NEWL F0F0F2F4D1C1D5C540			1433022	
D	101345	75012:0003	2P- MO EXPERTISE	AJ	M CH	MA S EMP-EXPERTISE	75012:0001.AO	75012:0001.AP			1433024	A
D	101345	75012:0001	2P- MO EMPLOYEE	AJ	O CH	S EMP-EXPERTISE	75012:0003.AN	75012:0004.BN			1433026	P
D	101345	75012:0004	2P+ MO EXPERTISE	DI	M CH	MA S EMP-EXPERTISE	75012:0001.BO	75012:0003.BN			1433028	A
D	101345	75012:0001	2P- MO EMPLOYEE	AJ	O CH	S EMP-EXPERTISE	75012:0004.AN	75012:0003.BN			1433030	P
D	101345	75012:0003	2P- MO EXPERTISE	AJ	M CH	MA S EMP-EXPERTISE	75012:0001.AO	75012:0004.AP			1433032	A
D	101345	75012:0004	2P+ MO EXPERTISE	CO	M CH	MA S EMP-EXPERTISE	75012:0001.AO	75012:0003.AN			1433034	A
D	101345	75163:0005	2D- MO SR8-INDX	AJ	A I	SY A SR8-INDEX-SET	75163:0001.AO	75163:0001.AP			1433036	Z
D	101345	75012:0004	2P+ MO EXPERTISE	DI	M IU	MA S SKILL-EXPERTISE	75163:0001.BO	75163:0005.BN			1433038	Q
D	101345	75163:0005	2D- MO SR8-INDX	AJ	A I	SY A SR8-INDEX-SET	75163:0001.AO	75163:0001.AP			1433042	Z
D	101345	75012:0004	2P+ MO EXPERTISE	CO	M IU	MA S SKILL-EXPERTISE	75163:0001.AO	75163:0005.AN			1433044	Q
D	101345	75012:0004	2D+ MO EXPERTISE				05720128	NEWL F0F5F7F2F0F1F2F840			1433046	
D	101345	75012:0001	3P- ST EMPLOYEE	CO	O CH	S EMP-EXPERTISE	75012:0006.AN	75012:0004.BN			1433080	P
D	101345	75012:0004	3P- ST EXPERTISE	CO	M CH	MA S EMP-EXPERTISE	75012:0001.AO	75012:0006.AP			1433082	A
D	101345	75163:0005	3D- ST SR8-INDX	CO	A I	SY A SR8-INDEX-SET	75163:0001.AO	75163:0001.AP			1433084	Z
D	101345	75012:0006	3D+ ST EXPERTISE				07012201	NEWL F0F7F9F4F0F6F2F840			1433085	
D	101345	75168:0006	4D+ MO SKILL				9170THINKING	AB F9F1F7F0E3C8C9D5D2			1433087	
D	101345	75160:0010	5P- DI DEPARTMENT	DI	O CH	S DEPT-EMPLOYEE	75012:0001.BN	75160:0010.AN			1433089	P
D	101345	75012:0001	5P+ DI EMPLOYEE	DI	M CH	OA S DEPT-EMPLOYEE	75160:0010.BO	75160:0010.BN			1433091	A
D	101345	75166:0001	6D+ MO DEPARTMENT				3100GLOBAL	SOFTWARE F3F1F0F0C7D3D6C2C1			1433093	
D	101345	75040:0001	7P- CO EMPLOYEE	CO	M CH	OA S DEPT-EMPLOYEE	75166:0001.AO	75012:0001.AN			1433097	A
D	101345	75007:0001	7P- CO EMPLOYEE	CO	M CH	OA S DEPT-EMPLOYEE	75166:0001.AO	75012:0001.AP			1433099	A
D	101345	75012:0001	7P+ CO EMPLOYEE	CO	M CH	OA S DEPT-EMPLOYEE	75166:0001.AO	75007:0001.AN			1433101	A
D	101345	75160:0010	8P- ST DEPARTMENT	CO	M CH	MA S CALC	75160:0011.AN	75160:0000.BN			1433104	P
D	101345	75160:0000	8P- ST SR1-CALC	CO	O CH	SY S SR1-REC-CALC-SET	75160:0011.AP	75160:0010.BP			1433106	P
D	101345	75154:0004	8D- ST SR8-INDX	CO	A I	SY A SR8-INDEX-SET	75154:0001.AO	75154:0002.AP			1433110	Z
D	101345	75160:0011	8D+ ST JOB				7777V.P.	GLOBAL SO F7F7F7F5E4BD74B40			1433111	
D	101345	75012:0001	9P- ST EMPLOYEE	CO	O CH	F EMP-EMPOSITION	75012:0007.AN	75012:0002.BN			1433145	P
D	101345	75012:0002	9P- ST EMPPOSITION	CO	M CH	MA F EMP-EMPOSITION	75012:0001.AO	75012:0007.AP			1433147	A
D	101345	75012:0007	9D+ ST EMPPOSITION				01220199123142@000	F9F4F0F6F2F8F9F9F1			1433148	
D	101345	75160:0011	0P- CO JOB	CO	O CH	OM N JOB-EMPOSITION	75012:0007.AN	75160:0011.BN			1433150	P
D	101345	75012:0007	0P+ CO EMPPOSITION	CO	M CH	OM N JOB-EMPOSITION	75160:0011.AO	75160:0011.AP			1433152	A
D	101345	75022:0001	1D+ MO EMPLOYEE				0003JENNIFER	GARF F0F0F0F3D1C5D5D5C9			1433154	
D	101345	75022:0016	2D+ EA STRUCTURE				A 820101	IIFER GARF C140F8F2F0F1F0F1C9			1433155	
D	101345	75022:0015	2P- EA STRUCTURE	DI	M CH	MA N MANAGES	75022:0001.BO	75022:0016.BP			1433157	A

Figure 37 • Sample Report 05 (page 2)

REPORT 05.	ALLEN SYSTEMS GROUP - TRACER				TRACER TRACK SET AND DATA RECORD UPDATES				RUN DATE	1/22/01	PAGE	2
T UPDATING	UPDATED-RECORDS	DDT VE	UPDATED-RECORD	AU R TY ST S	CHANGED-SET-NAME	SET-POINTER.WW	SET-POINTER.WW	SEQUENCE-#	S			
L RUN-UNIT#	DBKEY-PAGE-LINE	IPG	RB	VB S ST DC O	OR-DATA-IN-CHAR.	IP	OR-DATA-IN-HEX	.IP	FROM-SRC-J	P		
D 101345	75022:0017	2P-	EA STRUCTURE	DI M CH MA N	MANAGES	75022:0001.BO	75022:0016.BN	1433159	A			
D 101345	75012:0001	2P-	EA EMPLOYEE	DI O CH N	REPORTS-TO	75022:0016.BN	75012:0001.AN	1433161	P			
D 101345	75040:0001	3P-	ST EMPLOYEE	CO M CH MA S	CALC	75040:0006.AN	75040:0000.BN	1433166	P			
D 101345	75040:0000	3P-	ST SR1-CALC	CO O CH SY S	SR1-REC-CALC-SET	75040:0006.AP	75040:0001.BP	1433168	P			
D 101345	75028:0005	3P-	ST EMPLOYEE	CO M CH OA S	DEPT-EMPLOYEE	75170:0001.AO	75040:0006.AN	1433170	A			
D 101345	75043:0006	3P-	ST EMPLOYEE	CO M CH OA S	DEPT-EMPLOYEE	75170:0001.AO	75040:0006.AP	1433172	A			
D 101345	75002:0004	3D-	ST SR8-INDX	CO A I SY A	SR8-INDEX-SET	75002:0001.AO	75002:0002.AP	1433176	Z			
D 101345	75174:0012	3D-	ST SR8-INDX	CO A I SY A	SR8-INDEX-SET	75174:0001.AO	75174:0001.AP	1433180	Z			
D 101345	75040:0006	3D+	ST EMPLOYEE			7777SELL	SOFT F7F7F7E2C5D3D340	1433181				
D 101345	75012:0001	4P-	ST EMPLOYEE	CO O CH N	MANAGES	75012:0008.AN	75012:0001.BN	1433215	P			
D 101345	75012:0008	4P+	ST STRUCTURE			A 012201	SOFT C140F9F4F0F6F2F840	1433216				
D 101345	75040:0006	5P-	CO EMPLOYEE	CO O CH N	REPORTS-TO	75012:0008.AN	75040:0006.BN	1433218	P			
D 101345	75012:0008	5P+	CO STRUCTURE	CO M CH OM N	REPORTS-TO	75040:0006.AO	75040:0006.AP	1433220	A			

This section describes each field in Report 05.

**T  
L**

Displays the categories of LINE TYPE. These are the valid values:

- B = Beginning of run unit information
- D = Detail information about a specific record occurrence update

**UPDATING  
RUN-UNIT#**

Displays the unique number IDMS assigns to the run unit that issued the database update. This column is filled in for either of the two line types.

**UPDATED-RECORDS  
DBKEY-PAGE-LINE**

Displays the dbkey of the record that was changed.

**D  
I**

Displays the DML indicator. The character in this column enables the reader to determine when the updates change from one DML verb to the next.

For example, if a CONNECT verb caused the record occurrences to be updated, the three lines on the report that are associated with that CONNECT verb would each have the same character in this position (e.g., 1). A different character (e.g., 2) would be in this column on the next line, showing the next record occurrence updated by the next DML update verb for this run unit (perhaps a different CONNECT verb).

**Note:**

When there are many concurrent run units simultaneously updating records, the value in this column may not change when the updating run unit number changes from line to line.

When reading the report, check this column to quickly determine whether the DML verb or the currently updating run unit has changed.

**D**

**P**

Displays the data or pointer-only code in this column, indicating that only one or more of the pointer dbkeys have changed, or that some of the data portion of the record has changed.

D = Indicates that some of the data portion of the record was changed

P = Indicates that none of the data was changed, but at least one of the dbkeys used to establish the record's logical position in one of its sets has changed

**T**

**G**

Indicates whether this record occurrence is the target record occurrence of the DML update verb, or one of the adjacent record occurrences in one of the sets in which the target record participates.

+ = Indicates that this record occurrence is the one the DML update verb was primarily supposed to act upon

- = Indicates that this record occurrence was automatically updated by IDMS because it is internally affected by the update verb

**VE**

**RB**

Displays the abbreviation for the update verb issued by the application program that caused this update. These are the available values:

ST = STORE

CO = CONNECT

DI = DISCONNECT

ER = ERASE

EA = ERASE ALL

EP = ERASE PERMANENT

ES = ERASE SELECTIVE

MO = MODIFY

**UPDATED-RECORD**

Name of the record type that was updated.

**AU**  
**VB**

Indicates the AUTOMATIC VERB used when IDMS must automatically disconnect or connect the target record occurrence from sets in which it participates. This indicates what type of operation is being done on this particular updated journal record.

For instance, suppose the update verb is MODIFY and one of the changed fields happens to be a sort key in a set to which this record was automatically connected. IDMS must disconnect the record from its old position in that set and subsequently connect the record in the same set, at the new appropriate logical position, based on the new value of the field that the set is sorted on.

This column contains DI on the line produced based on the journal record whose set pointers were adjusted to disconnect the record from its old logical position in the set. It will contain CO for the subsequent connect to put the record in its new position in the set based on the new value of the sort field.

**R**  
**S**

Indicates ROLE IN SET, showing whether the updated record occurrence participates as an owner (O), or a member (M) in the set whose pointer(s) were updated.

**TY**  
**ST**

Indicates TYPE OF SET, showing the type of set that was updated. These are the valid values for this column:

- CH = Chained Set
- IS = Indexed Set System Owned (integrated index with SR7 owner)
- IU = Indexed Set User Owned (integrated index with regular owner)

**ST**  
**DC**

Indicates Set Disconnect and Connect settings, showing how the schema settings for disconnect and connect verbs have been specified for the set that has been updated.

For instance, a common setting is MA (Mandatory Automatic), meaning that member record occurrences in this set may only be disconnected through an ERASE verb. When record occurrences of the member record type in the set are stored, IDMS automatically connects them at the appropriate logical position in the set. The position of this record occurrence in this set is based on currency and/or the value in the field(s) serving as a sort key (if any) at the time of the STORE.

**S  
O**

Indicates Set Order, showing how the set is ordered. These are the valid values:

S	=	Sorted
N	=	Next
P	=	Prior
F	=	First
L	=	Last

**CHANGED-SET-NAME**

Indicates Set Name, showing the name of the set whose pointer(s) changed.

**SET-POINTER.WW  
OR DATA IN CHAR.IP**

Displays dbkeys extracted from the pointer positions of the updated record occurrence associated with the set that has been affected by the update. If the update affected data in the updated record occurrence, these fields contain the first 18 bytes of the record in displayable (non-hex) form, and the first nine bytes of the record in hexadecimal printable characters.

If only pointers in the set were updated (no data), this column contains a dbkey (page and line index), followed by a period (separator) and two indicators. The two columns following the period provide information about the dbkey displayed to the left.

The first indicator tells which image, where:

B	Indicates the before image of the dbkey
A	Indicates the after image

The second indicator tells which pointer, where:

N	=	NEXT
P	=	PRIOR
O	=	OWNER

This description also applies to the next column where another dbkey from the updated set's pointers will be displayed. If the updated set has owner pointers, or if the set is an indexed set where owner pointers were not specified in the schema, this first SET-POINTER column contains the dbkey of the OWNER of the changed set. For indexed sets that did not specify owner pointers in the schema, Tracer gets the OWNER dbkey of the set from the most recently journaled SR8 record and places it here.

If some of the updated record data was changed, this column contains the first 18 bytes of the record. On a MODIFY, Tracer uses the after image of the updated record. If a particular character in the 18 bytes is non-printable, the @ character is placed in that position.

**SET-POINTER.WW  
OR-DATA-IN-HEX.IP**

Contains the first nine bytes of the record printed in hexadecimal if some of the updated record's data was changed.

When a dbkey is displayed, see the description above for the meaning of the indicators to the right of the period.

Note the two Set-Pointer columns: Tracer attempts to tell you which dbkey could re-establish appropriate currency if you wrote a program to do a rollback operation to undo the effects of a program that earlier updated the IDMS database.

**SEQUENCE-#  
FROM-SRC-J**

Indicates the sequence number of the source archived journal record.

**S  
P**

Set Pointers contain an indicator that tells which pointers the set contains. These are the valid values for this column:

- A = Next, prior and owner pointers
- N = Only next pointers
- P = Next and prior pointers
- Q = Owner and next pointers
- Z = Next, prior, and owner pointers on SR8 record

The main purpose of Report 05 is to provide additional information about updates done by either a CONNECT or a DISCONNECT verb and the pointers that are automatically changed by IDMS in set relationships.

### **Report 11 - Track Physical Record Updates (Record Level)**

Report 11 lists one line for each record (or matched pair of records) found on the reformatted journal file. It is basically an enhanced version of Report 01. By setting various Tracer input parameters, you can change the look of Report 11.

Report 11 enables you to:

- Print all dbkey set pointers for every journal record, or print only those that changed.
- Hex-dump the entire record, any range of bytes, or only the control keys. You may also highlight the bytes that changed.
- Print a report banner and a key to abbreviation footers.
- Use the reformatted journal file as input.

Report 11 can be produced from the reformatted journal by running the DBTRRFMT program, whereas Report 01 can be created only by executing the DBTRFMT program.

**SORT SEQUENCE:** Date/Time (the order records are stored on the IDMS journal).





**TASK ID**

Displays the unique number that IDMS assigns to each task when the updating run unit was from an internal online environment. Task ID is BATCH for batch jobs.

**DBKEY**

**PAGE**

Together with DBKY/LNIX, indicates the dbkey of the record that was updated. On COMT, ENDJ, and ABRT records, this column shows the number of physical pages written by IDMS.

**DBKEY**

**LNIX**

Together with DBKEY/PAGE, identifies the dbkey of the record that was updated.

**PROGRAM**

**NAME**

Indicates the name that was in the PROGRAM-NAME field of the SUBSCHEMA-CTRL communication area when the program issued the BIND RUN UNIT verb.

**D**

**P**

Code that indicates whether the data portion or only the pointer portion in this record changed. These are the valid values:

- P = Only pointer changes were made on this record
- D = Indicates that the data portion of this record was journaled due to an update

**VE**

**RB**

Indicates the abbreviation for the update verb issued by the application program that caused this update. These are the valid values:

- ST = STORE
- CO = CONNECT
- DI = DISCONNECT
- ER = ERASE
- EA = ERASE ALL
- EP = ERASE PERMANENT
- ES = ERASE SELECTIVE
- MO = MODIFY

**RECORD  
NAME**

Name of the record type that was updated.

**USERID  
IF X14**

Displays the first eight characters of the user ID associated with the run unit that caused the update. You must have installed the IDMS System Exit 14 (RHDCUX14) in order for this field to be set.

**RFMT-RC#**

Indicates the sequence number on the DBTRFMT file.

**CHGD  
OFST**

Displays the position in the record occurrence of the first data character that was changed. It is only set to a non-zero value when the update verb is MODIFY.

**JRNL-SEQNC  
NUMBER**

Shows the unique journal sequence number assigned to SEQ NUMBER for this journal record. Can be used to match an update from this summary report to a Tracker report. To display this column, you must specify this input parameter:

RECRPT-COLUMNS            165

**BFOR DATA  
1234567890**

Displays the first 10-20 bytes of the data portion of the modified record when this line was built from a BFOR journal record. (Only applicable when the verb is MODIFY or DELETE.) To display 20 bytes, you must specify this input parameter:

RECRPT-COLUMNS            165

**AFTR DATA  
1234567890**

Displays the first 10-20 bytes of the data portion of the modified record when this line was built from an AFTR journal record. (Only applicable when the verb is MODIFY or STORE.) To display 20 bytes, you must specify this input parameter:

RECRPT-COLUMNS            165

**PAGE  
GRUP**

Displays the page group number to make a dbkey unique in shops that have segmented databases or that reuse page ranges when defining their database areas.

## Physical Record Updates - Fundamental Differences

The reformatted journal file does not contain this; therefore, they are not included on Report 11:

- SR0-SYMMETRY records
- BG (Begin), EJ (Endjob), AB (Abort), CO (Checkpoint)
- Run Unit Statistics.

You may optionally include pointer-only records and the SR7/SR8 records in the reformatted journal file when running the DBTRFMT utility.

These parameters determine whether or not these two record types are included in the reformatted journal file.

### **SEL-YN-RETURN-POINTERS = Y**

Determines whether Pointer-Only records are included in the reformatted journal file (default is YES). This parameter must be set to Y in order to include pointer-only records on Report 11 or on the Tracker report.

### **SEL-YN-RETURN-SYSTEM-RECORDS = Y**

Tells Tracer whether or not to return the SR7/SR8 records.

### **SEL-WRITE-SYSTEM-RECS-RFMT = Y**

Tells Tracer whether or not to write system records to the reformatted journal file. System records include:

- SR1 - Calc Chain
- SR7/SR8 - Indexes

## Database (Tuning and Statistics) Reports

Tracer database reports are produced by the DBTDBART program and are described in this section. Details about each report and its individual fields are defined and illustrated in numeric order.

Tracer produces these tuning and statistics reports:

**DBTRPT30.** Concurrent Run Units Detail Report

**DBTRPT31.** Run Units Associated with Record Locks Report

**DBTRPT33.** User Report by Application

**DBTRPT34.** Run Unit Reports

**DBTRPT37.** Concurrent Run Unit Summary Reports

**DBTRPT38.** Program Resource Reports (reports 39, 40, and, 41)

**DBTRPT39.** Program Resource Report by Program Name

**DBTRPT40.** Program Resource Report by Most Used

**DBTRPT41.** Program Resource Report by Longest Run

**DBTFIGHT.** Contention and Record Locking Report

**Tracer Summary.** Record Update Counts and Totals

If you wish to write your own performance or debug reports, you may run them using DBTDBART as your driver and use any of the supplied reports as guidelines. Report numbers 80 through 99 are reserved for user-defined reports.

### ***DBTRPT30 - Tracer Concurrent Run Unit Detail Report***

The Concurrent Run Unit Detail report is designed to help you determine the run units that are not performing at their best. Performance is measured by aborts or the actual elapsed time compared to the time that you consider adequate for your environment.

If the reason for the run unit taking too long to run is contention with another run unit, then this report can help you determine which run unit was in contention. Tracer reports on every run unit that was simultaneously active with your run unit.

This report displays, in two lines, all the information accumulated at the end of the job run unit. For each ending or aborting run unit, two additional lines display concurrent run units. Each set of ending run units are separated by a blank line. You invoke this report by specifying DBTRPT30 as control card in the DBTDBART program.

[Figure 40](#) shows a sample of Tracer's Report 30.

Figure 40 • Sample Report 30

REPORT 30.													TRACER CONCURRENT RUN UNIT DETAIL REPORT				DATE	1/22/01	PAGE	1		
C A	RUN	UNIT	PROGRAM	TASK ID	-----STARTING-----	ELAPS	CKECK	PT	RECOV	MAX	RC	USER	ID	SUBSCHEMA								
G V	NUMBER	NAME	NUMBER	NUMBER	DATE	TIME	TIME	TIME	TIME	TIME	TIME	ID	NAME									
													-----RECORDS-----	PAGE/LINE	PAGES	AREA	RECS	COMT	EXCLU	SELEC	BYTES	
													NAME	ID	NUMBER	READ	WROTE	READY	UPDATE	LOCKS	LOCKS	JRNL
E D	114392	DBOLMAIN	480	01/20/01	20.29.00.71	60080	20.39.01.51	60080	60080	ASG1	EMPSS01											
							440	:	36	7	3	18				9	4	3352				
E D	114410	DBOLMAIN	495	01/20/01	20.39.24.10	10505	20.41.09.15	10505	10505	ASG2	EMPSS01											
							420	:	1	1	3	4				4	2	1412				
E D	114421	DBOLMAIN	506	01/20/01	20.43.14.06	10026	20.44.54.32	10026	10026	ASG2	EMPSS01											
							440	:	2	1	3	1				2	1	804				
E B	114387	EMPLPOPU	2027537674	01/20/01	20.27.57.74	1382	20.28.11.56	1382	1382	TSO011P1	EMPSS01											
							415	:	472	61	3	199				105	2	84800				
E B	114415	EMPLPOPU	2041385651	01/20/01	20.41.38.57	1111	20.41.49.68	1111	1111	TSO011P2	EMPSS01											
							415	:	394	14	3	36				36	2	15704				

This section describes each field in Report 30.

**REPORT 30**

Indicates the current report number of the Tracer Concurrent Run Unit Detail report.

**DATE**

Displays the month, day, and year when this report was generated.

**PAGE**

Displays the sequential page number of this report.

These are the report's column headings:

**C  
G**

Represents the abend type. These are the valid values:

- A = Abort
- E = End of Job
- B = Begin
- C = Commit

**A**  
**V**

Represents the environment code for this run unit. It tells you whether the run unit was initiated by a program which was loaded into a IDMS program pool, or another region in the operating system (online or batch). This one-byte code is the first digit of the first four bytes of the local ID internal variable subschema field. These are the valid values:

D = DBDC  
B = Batch  
C = CICS  
T = TSO

**RUN UNIT  
NUMBER**

Displays the unique number that IDMS assigns to each run unit.

**PROGRAM  
NAME**

Displays the name which was in the PROGRAM-NAME field of the SUBSCHEMA CTRL communication area when the program issued the BIND RUN UNIT verb.

**TASK ID  
NUMBER**

Displays the unique number that IDMS assigns to each task when the updating run unit was from an internal online environment. This is 0 if the run unit was from an external environment (i.e., something other than the IDMS-CV address space).

**STARTING  
DATE**

Displays the month, day, and year on the BGIN run unit for the run unit that caused this update.

**STARTING  
TIME**

Displays the hour, minute, second, and hundredths-of-seconds which were on the BGIN run unit for the run unit which caused this update.

**ELAPS  
TIME**

Displays the total elapsed time for the major run unit.

**CHECK PT  
TIME**

Indicates the last check point hour, minute, second, and hundredths-of-seconds for this run unit.

**RECOV  
TIME**

Displays the total elapsed time for the run unit, or the elapsed time for the recovery unit of the concurrent program, at the time when this run unit ended (depending on the line type).

**MAX RC**

Displays the largest elapsed time for any recovery unit in hours, minutes, seconds, and hundredths-of-seconds. For this report, a recovery unit is defined as the time between a BGIN and the first COMT, ENDJ, or ABRT journal record, or the time between one COMT record and the next COMT, ENDJ, or ABRT record. For online internal run units, there may be only one recovery unit, (between the BGIN and ENDJ or ABRT records) since a COMT is never issued.

**USER ID**

Displays the first eight characters of the user ID associated with the run unit which caused the update. You must have installed the CA-IDMS System Exit 14 (RHDCUX14) in order for this field to be set.

**SUBSCHEMA NAME**

If the CA-IDMS User Exit 14 was active, the subschema name that was used for this run unit is displayed.

These are descriptions of the second line of Report 30 shown in [Figure 40 on page 132](#):

**RECORDS  
NAME**

Displays the record name for the last record that was updated by this run unit. The record name does not display if this Record ID is not defined to the subschema being used.

**RECORDS  
ID**

Displays the record ID for the last record that was updated by this run unit.

**PAGE/LINE  
NUMBER**

Displays the page and line number of the last record ID as per the IDMS journal archive file.

**PAGES  
READ**

Displays the number of pages read for this run unit.

**PAGES  
WROTE**

Displays the number of pages written for this run unit.

**AREA  
READY**

Displays the number of areas that were readied by this run unit.

**RECS  
UPDATE**

Displays the number of records that were updated by this run unit.

**COMT**

Displays the number of commit records that were processed by this run unit.

**EXCLU  
LOCKS**

Displays the number of exclusive locks that were issued by this run unit.

**SELEC  
LOCKS**

Displays the number of selective locks that were issued by this run unit.

**BYTES  
JRNL**

Displays the number of IDMS journal bytes that this run unit used through the last check point.

***DBTRPT31 - Run Units Associated with Record Locks***

The Tracer Record Locks report is designed to help you determine which run units Tracer suspects to be related to record locks. This report displays, in two lines, all of the information about these types of run units. The fields in columns 84 through 132 (beginning with SUSPECT PROGRAM field) have a different meaning, depending on whether they display on a primary line or on one of its concurrent lines. A primary line always contains a value in the TP column, while its concurrent line(s) have a blank space in the TP column.

On the primary line, Tracer reports on the period when a run unit was waiting on a record occurrence locked by a concurrent update run unit. This run unit waited longer than the specified TR-I-HUND-SECS-WAIT-B4-COMT time. The wait occurred either between the BGIN record and the first update, or between the first update this run unit issued after the wait occurred and immediately after a concurrent run unit produced a checkpoint record. Concurrent line fields report information about run units that were still active when this (primary) run unit finished.

[Figure 41](#) shows a sample of Tracer's Report 31. You invoke this report by specifying DBTRPT31 as control card in the DBTDBART program.

Figure 41 • Sample Report 31

```

REPORT 31.                TRACER - RUN UNITS SUSPECTED TO BE ASSOCIATED WITH RECORD LOCKS                DATE 1/22/01 PAGE      1
T E ELAPSED PROGRAM RUN-UNIT-# TOTAL  PAGES MX-TIME #RECS STARTD S-DT MX-ELAP IXU SUSPECT CONTENTION WAITED V RCRD S-LOCKED-DBKY I
P V TIME/RU NAME                PHY-IO WRITN BTWN-UP UPDTD HMMSS MMDD TIM-RCU DML PROGRAM RUN-UNIT-# TIME-C B B ID-# PAGE-NUMB:LNK X
-----
*-----
L D  600.80 DBOLMAIN    114392    43    7 8103.00    12 202900 0120    600.80
*-----
L D  105.05 DBOLMAIN    114410     2    1  72.00     8 203924 0120    105.05
*-----
L D  100.26 DBOLMAIN    114421     3    1          1 204314 0120    100.26
*-----
L B  13.82 EMPLPOPU    114387    533   61 120.00   389 202757 0120    13.82  7
*-----
L B  11.11 EMPLPOPU    114415    408   14  46.00    65 204138 0120    11.11  4
    
```

This section describes each field in Report 31.

**REPORT 31**

Indicates the current report number of the Tracer Run Units Associated with Record Locks report.

**DATE**

Displays the month, day, and year when this report was generated.

**PAGE**

Displays the sequential page number of this report.

**T  
P**

Represents the type of run unit that Tracer suspects caused the record lock. These are the valid values, in the order in which the report is sorted:

- X = Update run unit that aborted
- W = Run units that have waited on another concurrently executing run unit
- L = Run units that ran longer than the specified time limit, in elapsed seconds
- Y = Caused another run unit to abort

For concurrent lines, the value Y indicates that Tracer suspects this primary run unit caused the abort. These fields are the same as described previously for primary lines, except that the waiting time may be less than TR-I-HUND-SECS-WAIT-B4-COMT. This is because Tracer knows that, in cases of deadlocks, a run unit may not wait long on the other deadlocks before aborting.

**Note:** \_\_\_\_\_

Tracer reports information on the primary run unit that updated a run unit after the concurrent run unit aborted. As a result, Y-type run units may be reported which did not actually terminate the other aborted run unit. The log data dump verifies the run unit type.

---

**E  
V**

Represents the environment code for this run unit. It tells you whether the run unit was initiated by a program which was loaded into a IDMS program pool, or another region in the operating system (online or batch). This one-byte code is the first digit of the first four bytes of the local ID internal variable subschema field. These are the valid values:

D = DBDC  
B = Batch  
C = CICS  
T = TSO

**ELAPSED  
TIME/RU**

Displays the total elapsed time for the major run unit (in wall-clock seconds and hundredths of seconds) that transpired between the time on the BGIN record and the ENDJ or ABRT journal records for the run unit.

**PROGRAM  
NAME**

Displays the name which was in the PROGRAM-NAME field in the SUBSCHEMA-CTRL communication area when the program issued the BIND RUN UNIT verb.

**RUN UNIT-#**

Displays the unique number that IDMS assigns to each run unit.

**TOTAL  
PHY-IO**

Displays the result of pages read and pages written. This includes all physical I/Os to database datasets done by this run unit, as recorded in the IDMS VB50 statistics.

**PAGES  
WRITN**

Displays the number of pages written (physical output) for this run unit.

**Note:** \_\_\_\_\_

Since IDMS flushes each updated buffer page to the database (and journal files) before control may be returned to the updating program, the buffers do not reduce the elapsed time on pages that must be written.

Because of the way IDMS determines this field, it is possible for it to be equal to zero even though an update actually did take place. This is unusual, but in very high throughput situations this might happen. (There is a CA documentation PTF about this.)

---

**MX-TIME  
BTWN-UP**

Displays the maximum time between updates in seconds and hundredths of seconds (SS.HS). This is the elapsed time between any two journaled images produced because of update DML verbs initiated by this run unit.

**#RECS**

Displays the number of records updated. This is the number of times IDMS updated record occurrences on behalf of this run unit.

**STARTD  
HHMMSS**

Displays the time the run unit was written on the BGIN journal record in hours, minutes, and seconds (*HH.MM.SS*).

**S-DT  
MMDD**

Started date of this run-unit in month and day.

**MX-ELAP  
TIM-RCU**

Displays the maximum elapsed time for any recovery unit for this run unit in seconds and hundreds of seconds. A recovery unit is the time between two consecutive journal records for a BGIN, COMT, ENDJ, or ABRT run unit record.

**IXU  
DML**

Displays the maximum index record occurrences updated by any DML verb. This is the maximum number of system maintained index type records (SR7 or SR8) that were updated during the processing of what Tracer assumed was a single DML update verb.

**SUSPECT  
PROGRAM**

Displays the name of the concurrent program for which Tracer suspects this run unit waited.

For concurrent lines, displays the name of the concurrent program that was still running when this run unit finished.

**CONTENTION  
RUN-UNIT-#**

Displays the run unit number of the concurrent program for which Tracer suspects this run unit waited.

For concurrent lines, displays the run unit number of the concurrent program that was still running when this run unit finished.

**WAITED  
TIME-C**

Displays the number of seconds and hundreds of seconds (SS.HS) that transpired during the period that this run unit did not do any updates and the point when the suspected contention run unit issued the CHECKPOINT verb.

**Note:** \_\_\_\_\_

The one instance when this value could be greater than MX-TIME BTWN-UP is when the wait time is less than the BGIN record time and greater than the first update DML verb time. The MX-TIME BTWN-UP field only considers time between actual update verbs.

This value takes into consideration the time between the run unit BIND and the first update DML verb.  
\_\_\_\_\_

For concurrent lines, displays the elapsed time between the last checkpoint and the finish of the primary run unit.

**V  
B**

Displays the type of update verb processed by this run unit immediately after the suspected contention run unit issued the CHECKPOINT verb.

For concurrent lines, displays the type of update verb processed by this concurrent run unit before the primary run unit finished.

**RCRD  
ID-#**

Displays the record ID of the record that this run unit updated immediately after the suspected contention run unit issued the CHECKPOINT verb.

For concurrent lines, displays the last record ID that this concurrent run unit updated before the primary run unit finished.

**S-LOCKED-DBKEY**

**PAGE-NUMB:LNK**

Displays the dbkey of the record occurrence that this run unit updated immediately after the suspected contention run unit issued the CHECKPOINT verb.

For concurrent lines, displays the last dbkey this run unit updated before the primary run unit finished.

**I**  
**X**

Displays a code indicating whether a system-maintained index record (i.e., SR7 or SR8) was updated by IDMS when processing the update DML verb issued by this run unit immediately after the suspected contention run unit issued the CHECKPOINT verb. An X in this column indicates that at least one index record was updated by the resuming DML verb; otherwise, the updated record did not require IDMS to update an index record.

**DBTRPT33 - Tracer User Report by Subschema**

Tracer's User Report by Subschema provides information regarding user access to the various application systems in your environment. If the Tracer-provided IDMS User Exit 14 is active, this report also shows the users that were associated with run units that issued a READY verb specifying an update access category. You invoke this report by specifying DBTRPT33 as control card in the DBTDBART program.

[Figure 42](#) shows a sample of Tracer's Report 33.

**Figure 42 • Sample Report 33**

REPORT 33.	TRACER USER REPORT BY SUBSCHEMA	DATE	01/20/01	PAGE	1
	SUBSCHEMA IS EMPSS01				
USER ID	TASKS	RUN UNITS	JRNL UNITS	USERS	
ASG1	1	1	1		
ASG2	4	4	2		
TSO011P1	2	1	1		
TSO011P2	2	1	1		
TOTALS	9	7	5	4	

This section describes each field in Report 33.

**REPORT 33**

The current report number.

**DATE**

The month, day, and year this report was generated.

**PAGE**

The sequential page number in this report.

**SUBSCHEMA**

The name of the subschema described by this report.

**USER ID**

The first eight characters of the user ID associated with the run unit that caused the update. You must have the IDMS System Exit 14 (RHDCUX14) installed at your site for this field to be set.

**TASKS**

The number of tasks that used this subschema.

**RUN UNITS**

The number of run units that used this subschema.

**JRNL UNITS**

The number of run units recorded on the IDMS journal by the ENDJ records.

**USERS**

The number of users that accessed the subschema.

**TOTALS**

Tallies tasks, run units, journal units, and users for the subschema and for all subschemas at End of Job.

***DBTRPT34 - Tracer Run Unit Report***

The Tracer Run Unit report provides run unit information on the IDMS journal archive tape. Each line on this report displays in the order in which the ENDJ or ABRT record displays on the IDMS journal archive tape—it is not sorted.

[Figure 43 on page 142](#) shows a sample of Report 34 in Tracer. You invoke this report by specifying DBTRPT34 as control card in the DBTDBART program.

Figure 43 • Sample Report 34

REPORT 34.		TRACER RUN-UNIT REPORT							DATE 1/22/01 PAGE 1			
H E U	START	TIME	RUN UNIT	PROGRAM	TASK ID	ELAPSED TIME	ELAPSED TIME	MX-INACT	NUMBER	MAX LOCKS	PAGES	PAGES
E V P	DATE	HR.MM.SS.HS		NAME	NUMBER	HR.MM.SS.HS	RECOVERY-UNIT	TIME-UD	COMMIT	RECOVERY-UNIT	WRITTEN	READ
E B U	012001	20.27.49.17	114386	RWFP0902		0.00.00.13	0.00.00.13	00.00.04		1	1	
E B U	012001	20.27.57.74	114387	EMFLPOFU		0.00.13.82	0.00.13.82	00.01.20		105	61	472
E B U	012001	20.28.49.57	114389	RWFP0902		0.00.00.36	0.00.00.36	00.00.05		1	1	
E B U	012001	20.29.50.41	114394	RWFP0902		0.00.00.10	0.00.00.10	00.00.00		1	1	
E B U	012001	20.30.50.78	114396	RWFP0902		0.00.00.14	0.00.00.14	00.00.00		1	1	
E B U	012001	20.31.51.14	114397	RWFP0902		0.00.00.10	0.00.00.10	00.00.00		1	1	
E B U	012001	20.32.51.49	114398	RWFP0902		0.00.00.10	0.00.00.10	00.00.00		1	1	
E B U	012001	20.33.51.80	114399	RWFP0902		0.00.00.10	0.00.00.10	00.00.00		1	1	
E B U	012001	20.34.52.14	114401	RWFP0902		0.00.00.08	0.00.00.08	00.00.00		1	1	
E B U	012001	20.35.52.45	114402	RWFP0902		0.00.00.11	0.00.00.11	00.00.00		1	1	
E B U	012001	20.36.53.05	114403	RWFP0902		0.00.00.09	0.00.00.09	00.00.00		1	1	
E B U	012001	20.37.53.37	114404	RWFP0902		0.00.00.11	0.00.00.11	00.00.00		1	1	
E B U	012001	20.38.53.70	114406	RWFP0902		0.00.00.11	0.00.00.11	00.00.04		1	1	
E D U	012001	20.29.00.71	114392	DBOLMAIN	480	0.10.00.80	0.10.00.80	03.01.03		9	7	36
E B U	012001	20.39.54.06	114412	RWFP0902		0.00.00.10	0.00.00.10	00.00.00		1	1	
E B U	012001	20.40.59.00	114414	RWFP0902		0.00.00.45	0.00.00.45	00.00.00		1	1	
E D U	012001	20.39.24.10	114410	DBOLMAIN	495	0.01.45.05	0.01.45.05	00.00.72		4	1	1
E B U	012001	20.41.38.57	114415	EMFLPOFU		0.00.11.11	0.00.11.11	00.00.46		36	14	394
E B U	012001	20.41.59.80	114416	RWFP0902		0.00.00.13	0.00.00.13	00.00.00		1	1	
E B U	012001	20.43.00.16	114418	RWFP0902		0.00.00.12	0.00.00.12	00.00.04		1	1	
E B U	012001	20.44.01.05	114424	RWFP0902		0.00.00.10	0.00.00.10	00.00.00		1	1	
E D U	012001	20.43.14.06	114421	DBOLMAIN	506	0.01.40.26	0.01.40.26	00.00.00		2	1 2	
E B U	012001	20.45.01.48	114425	RWFP0902		0.00.00.15	0.00.00.15	00.00.05		1	1	

This section describes each field in Report 34.

**REPORT 34**

The current report number.

**DATE**

The month, day, and year that this report was generated.

**PAGE**

The sequential page number in this report.

**H  
E**

Represents the How-Ended code for this run unit. These are the valid values:

A = Abort

E = End

**E  
V**

Represents the environment code for this run unit. It tells you whether the run unit was initiated by a program which was loaded in a IDMS program pool, or another region in the operating system (online or batch). This one-byte code is the first digit of the first four bytes of the Local ID internal variable subschema field. These are the valid values:

D = DBDC  
B = BATC  
C = CICS

**U  
P**

Displays the type of Ready (Update or Commit) that was issued by this run unit. These are the valid values:

U = Update  
C = Commit

**START  
DATE**

The month, day, and year which was on the BGIN run unit for the run unit that caused this update.

**START TIME  
HR.MM.SS.HS**

Displays the hour, minute, second, and hundredth-of-a second that the update occurred.

**RUN UNIT**

Displays the unique number that IDMS assigns to each run unit.

**PROGRAM  
NAME**

Displays the name which was in the PROGRAM-NAME field of the SUBSCHEMA-CTRL communication area when the program issued the BIND UNIT RUN verb.

**TASK ID  
NUMBER**

Displays the unique number that IDMS assigns to each task when the updating run unit was from an internal online environment. This is 0 if the run unit was from an external environment (i.e., something other than the IDMS-CV address space).

**ELAPSED TIME  
HR.MM.SS.HS**

Displays the elapsed time for the run unit in hours, minutes, seconds, and hundredths-of-seconds. In most cases this will be very close to the portion of response time for online transactions due to the IDMS-CV internal processing. If ADS/O was involved in the run unit's processing, depending on how the application is coded, it is possible to generate more than one run unit with a single press of the Enter key (or the Response key).

This is one reason why the task ID number is also on this report. The task ID number will be the same and the run unit number different in these situations. The sum of the elapsed times for all the run units for that task ID number would be closer to the portion of the response time associated with the IDMS-CV processing.

**ELAPSED TIME  
RECOVERY-UNIT**

Displays the largest elapsed time (in hours, minutes, seconds, and hundredths of seconds) for any recovery unit in this run unit. For this report, a recovery unit is defined as the time between a BGIN and the first COMT, ENDJ, or ABRT journal record or the time between one COMT record and the next COMT, ENDJ or ABRT record.

Usually, an online internal run unit does not issue the COMMIT verb. In this condition, there is only one recovery unit (between the BGIN and ENDJ or ABRT records).

**MX-INACT  
TIME-UD**

Displays the maximum inactive time between updates in seconds and hundredths of seconds. This is the time elapsed between any two journaled images produced because of update DML verbs initiated by this run unit.

**NUMBER  
COMMIT**

Displays the number of commits performed by IDMSIDMS for this run unit.

**MAX LOCKS  
RECOVERY-UNIT**

Displays the maximum number of update locks for any recovery unit in this run unit.

**PAGES  
WRITTEN**

Displays the number of pages that were physically written for the record that was updated to the database.

**PAGES  
READ**

Displays the number of pages read from the database.

## DBTRPT37 - Tracer Concurrent Run Units Summary Report

DBTRPT37 is designed to help you determine which run units are not effectively performing due to an abort or due to elapse time being greater than what your site considers adequate for the environment.

If run units are taking too long to run because they are in contention with other run units, this report helps you determine which run unit(s) are in contention. Each run unit that was active when this long-running or aborted run unit finished is listed on the report. (Possible causes of run units taking too long to run are because both are trying to update the same dbkey: possible deadlock or WAIT-ON-DBKEY.)

You invoke this report by specifying DBTRPT37 as control card in the DBTDBART program.

[Figure 44](#) shows a sample of Tracer's Report 37. The first line in the set is the run unit that either aborted or took too long. All other lines in the set are run units which were running at the same time. Each set is separated by a blank line.

**Figure 44 • Sample Report 37**

REPORT 37.													TRACER HIGH ELAPSED TIME WITH CONCURRENT RUN UNITS				DATE	1/22/01	PAGE	1						
H E U	START	START TIME	RUN UNIT	PROGRAM	TASK ID	ELAPSED TIME	ELAPSED TIME	NUMBER	MAX LOCKS	PAGE	SUB-	USER ID	E V P	DATE	HR.MM.SS.HS	NAME	NUMBER	HR.MM.SS.HS	RECOVERY-UNIT	COMMIT	RECOVERY-UNIT	WRITTEN	SCHEMA	EXIT	14	
E D U	012001	20.29.00.71	114392	DBOLMAIN	480	0.10.00.80	0.10.00.80		9	7	EMPSS01	ASG1														
E D U	012001	20.39.24.10	114410	DBOLMAIN	495	0.01.45.05	0.01.45.05		4	1	EMPSS01	ASG2														
E D U	012001	20.43.14.06	114421	DBOLMAIN	506	0.01.40.26	0.01.40.26		2	1	EMPSS01	ASG2														
E B U	012001	20.27.57.74	114387	EMPLPOFU		0.00.13.82	0.00.13.82		105	61	EMPSS01	TSO011P1														
E B U	012001	20.41.38.57	114415	EMPLPOFU		0.00.11.11	0.00.11.11		36	14	EMPSS01	TSO011P2														

This section describes each field in Report 37.

### REPORT 37

The current report number.

### DATE

The month, day, and year that this report was generated.

### PAGE

The sequential page number in this report.

**H  
E**

Represents the How-Ended code for this run unit. These are the valid values:

A = Abort

E = End

**E  
V**

Represents the environment code for this run unit and tells you whether it was initiated by a program that was loaded in a IDMS program pool, or another region in the operating system (online or batch). This one-byte code is the first digit of the first four bytes of the local ID internal variable subschema field. These are the available values:

D = DBCD

B = BATC

C = CICS

**U  
P**

Indicates the type of ready issued by the updating program. These are the available values:

U = Update

R = Retrieval

**START  
DATE**

Displays the month, day, and year for the run unit which caused this update.

**START TIME  
HR.MM.SS.HS**

Displays the hour, minute, second, and hundredth-of-a second that the update occurred.

**RUN UNIT**

Displays the unique number that IDMS assigns to each run unit.

**PROGRAM  
NAME**

Displays the name which was in the PROGRAM-NAME field of the SUBSCHEMA-CTRL communication area when the program issued the BIND RUN UNIT verb.

**TASK ID  
NUMBER**

Displays the unique number that IDMS assigns to each task when the updating run unit is from an internal online environment. This is 0 if the run unit is from an external environment (i.e., something other than the IDMS CV address space.)

**ELAPSED TIME  
HR.MM.SS.HS**

Displays the elapsed time for the run unit in hours, minutes, seconds, and hundredths-of-seconds. In most cases, this is very close to the portion of response time for online transactions due to the IDMS CV internal processing. If ADS/O was involved in the run unit's processing, depending on how the application is coded, it is possible to generate more than one run unit each time the Enter key (or the Response key) is pressed.

This is one reason why the task ID number is also on this report. The task ID number is the same and the run unit number is different in these situations. The sum of the elapsed times for all the run units for that task ID number would be closer to the portion of the response time associated with the IDMS CV processing.

**ELAPSED TIME  
RECOVERY-UNIT**

Displays the largest elapsed time for any recovery unit in this run unit in hours, minutes, seconds, and hundredths-of-seconds. For this report, a recovery unit is defined as the time between a BGIN and the first COMT, ENDJ, or ABRT journal record or the time between one COMT record and the next COMT, ENDJ, or ABRT record. Usually, an online internal run unit does not issue the COMMIT verb. In this condition, there is only one recovery unit (between the BGIN and ENDJ or ABRT records).

**NUMBER  
COMMIT**

Displays the number of commits performed by IDMS for this run unit.

**MAX LOCKS  
RECOVERY-UNIT**

Displays the maximum number of update locks for any recovery unit in this run unit.

**PAGE  
WRITTEN**

Displays the number of pages written by IDMS for this run unit.

**SUBSCHEMA**

Displays the subschema name which was used on the BIND RUN UNIT. You must have the IDMS System Exit 14 (RHDCUX14) installed at your site for this field to be set.

**USER ID  
EXIT 14**

Displays the first eight characters of the user ID associated with the run unit which caused the update.

You must have installed the IDMS System Exit 14 (RHDCUX14) for this field to be set.

***DBTRPT38 - Tracer Run Unit Resource Report***

The Tracer Run Unit Resource Report accumulates the resources that a program used during the day and determines the averages and totals for that program.

The report is printed in three sequences:

- Program names in alphabetic order.
- Most used programs in descending order.
- Longest running programs in descending order.

For each program, the first line of the report contains the averages, and the second line contains the totals.

[Figure 45 on page 149](#) shows a sample of Tracer's Report 38. You invoke this report by specifying DBTRPT38 as control cards in the DBTDBART program.

Figure 45 • Sample Report 38

REPORT 38. TRACER RUN UNIT RESOURCE REPORT												DATE	1/22/01	PAGE	1
REPORT IN SEQUENCE BY PROGRAM NAME															
ENV	PROGRAM NAME	TIMES USED	ELAPSED SECONDS	JOURNALED BYTES	TOTAL I-O	PAGES WRITTEN	RECORDS REQUESTED	RECORDS UPDATED	NUMBER COMMITS	JOB ABRT	ABRT SUSP	FRAGMENT STORED			
DBDC	DBOLMAIN	3	268.70 806.11	1856 5568	16 48	3 9	20 60	7 21	0 0	0 0	0 0	0 0			
BATC	EMPLPOFU	2	12.46 24.93	50252 100504	470 941	37 75	855 1711	227 454	0 0	0 0	0 0	0 0			
BATC	REWP0902	18	0.14 2.58	554 9980	1 18	1 18	7 126	1 18	0 0	0 0	0 0	0 0			

REPORT 38. TRACER RUN UNIT RESOURCE REPORT												DATE	1/22/01	PAGE	1
REPORT IN SEQUENCE BY MOST USED															
ENV	PROGRAM NAME	TIMES USED	ELAPSED SECONDS	JOURNALED BYTES	TOTAL I-O	PAGES WRITTEN	RECORDS REQUESTED	RECORDS UPDATED	NUMBER COMMITS	JOB ABRT	ABRT SUSP	FRAGMENT STORED			
BATC	REWP0902	18	0.14 2.58	554 9980	1 18	1 18	7 126	1 18	0 0	0 0	0 0	0 0			
DBDC	DBOLMAIN	3	268.70 806.11	1856 5568	16 48	3 9	20 60	7 21	0 0	0 0	0 0	0 0			
BATC	EMPLPOFU	2	12.46 24.93	50252 100504	470 941	37 75	855 1711	227 454	0 0	0 0	0 0	0 0			

REPORT 38. DBTRACE RUN UNIT RESOURCE REPORT												DATE	1/22/01	PAGE	1
REPORT IN SEQUENCE BY LONGEST RUNNING															
ENV	PROGRAM NAME	TIMES USED	ELAPSED SECONDS	JOURNALED BYTES	TOTAL I-O	PAGES WRITTEN	RECORDS REQUESTED	RECORDS UPDATED	NUMBER COMMITS	JOB ABRT	ABRT SUSP	FRAGMENT STORED			
DBDC	DBOLMAIN	3	268.70 806.11	1856 5568	16 48	3 9	20 60	7 21	0 0	0 0	0 0	0 0			
BATC	EMPLPOFU	2	12.46 24.93	50252 100504	470 941	37 75	855 1711	227 454	0 0	0 0	0 0	0 0			
BATC	REWP0902	18	0.14 2.58	554 9980	1 18	1 18	7 126	1 18	0 0	0 0	0 0	0 0			

This section describes each field in Report 38.

### REPORT 38

The current report number.

### DATE

The month, day, and year that this report was generated.

### PAGE

The sequential page number in this report.

**ENV**

Represents the environment that this program ran under. These are the available values:

DBDC	The program was loaded into a IDMS program pool.
BATC	The program was loaded into an operating system address space other than that of the CV or CICS. This is a batch job that used the CV to access the database.

**PROGRAM  
NAME**

Displays the name of the program that was used for a run unit.

**TIMES  
USED**

Displays the total number of run units that used this program name when the BIND RUN UNIT command was issued.

**ELAPSED  
SECONDS**

Displays the average number of seconds a program used or the total number of seconds in hundreds of seconds.

**JOURNALED  
BYTES**

Displays the average number of bytes that went to the IDMS journal for a run unit.

**TOTAL  
I-O**

Displays the average number of pages written and read.

**PAGES  
WRITTEN**

Determines the average number of pages written.

**RECORDS  
REQUESTED**

Displays the average number of records requested.

**RECORDS  
UPDATED**

Displays the average number of records updated.

**NUMBER  
COMMITTS**

Displays the average number of commits.

**JOBS  
ABRT**

Displays the average number of times that a run unit aborted.

**ABRT  
SUSP**

Displays the average number of times that this run unit is suspected to have caused another run unit to abort.

**FRAGMENT  
STORED**

Displays the average number of stored fragmented records on the IDMS journal archive file.

**DBTRPT39-41 - Individual Run Unit Resource Reports**

If you want to produce the individual run unit resource reports of DBTRPT38, you may do so by specifying reports 39, 40, or 41, as applicable.

**DBTRPT39 - Program Resource by Program Name**

[Figure 46](#) shows a sample of Tracer's Report 39. DBTRPT39 produces the Tracer Run Unit Resource report in sequence by program name.

**Figure 46 • Sample Report 39**

REPORT 39.		TRACER RUN UNIT RESOURCE REPORT							DATE	1/22/01	PAGE	1
REPORT IN SEQUENCE BY PROGRAM NAME												
ENV	PROGRAM NAME	TIMES USED	ELAPSED SECONDS	JOURNALED BYTES	TOTAL I-O	PAGES WRITTEN	RECORDS REQUESTED	RECORDS UPDATED	NUMBER COMMITS	JOBS ABRT	ABRT SUSP	FRAGMENT STORED
DBDC	DBOLMAIN	3	268.70 806.11	1856 5568	16 48	3 9	20 60	7 21	0 0	0 0	0 0	0 0
BATC	EMPLPOFU	2	12.46 24.93	50252 100504	470 941	37 75	855 1711	227 454	0 0	0 0	0 0	0 0
BATC	REWP0902	18	0.14 2.58	554 9980	1 18	1 18	7 126	1 18	0 0	0 0	0 0	0 0

**DBTRPT40 - Program Resource Report by Most Used**

Figure 47 shows a sample of Tracer's Report 40. DBTRPT40 produces the Tracer Run Unit Resource report in sequence by most used.

**Figure 47 • Sample Report 40**

REPORT 40.		TRACER RUN UNIT RESOURCE REPORT										DATE	1/22/01	PAGE	1
REPORT IN SEQUENCE BY MOST USED															
ENV	PROGRAM NAME	TIMES USED	ELAPSED SECONDS	JOURNALED BYTES	TOTAL I-O	PAGES WRITTEN	RECORDS REQUESTED	RECORDS UPDATED	NUMBER COMMITS	JOB ABRT	ABRT SUSP	FRAGMENT STORED			
BATC	REWP0902		0.14	554	1	1	7	1	0	0	0	0			
		18	2.58	9980	18	18	126	18	0	0	0	0			
DBDC	DBOLMAIN		268.70	1856	16	3	20	7	0	0	0	0			
		3	806.11	5568	48	9	60	21	0	0	0	0			
BATC	EMPLPOFU		12.46	50252	470	37	855	227	0	0	0	0			
		2	24.93	100504	941	75	1711	454	0	0	0	0			

**DBTRPT41 - Program Resource Report by Longest Run**

Figure 48 shows a sample of Tracer's Report 41. DBTRPT41 produces the Tracer Run Unit Resource report in sequence by longest running.

**Figure 48 • Sample Report 41**

REPORT 41.		TRACER RUN UNIT RESOURCE REPORT										DATE	1/22/01	PAGE	1
REPORT IN SEQUENCE BY LONGEST RUNNING															
ENV	PROGRAM NAME	TIMES USED	ELAPSED SECONDS	JOURNALED BYTES	TOTAL I-O	PAGES WRITTEN	RECORDS REQUESTED	RECORDS UPDATED	NUMBER COMMITS	JOB ABRT	ABRT SUSP	FRAGMENT STORED			
DBDC	DBOLMAIN		268.70	1856	16	3	20	7	0	0	0	0			
		3	806.11	5568	48	9	60	21	0	0	0	0			
BATC	EMPLPOFU		12.46	50252	470	37	855	227	0	0	0	0			
		2	24.93	100504	941	75	1711	454	0	0	0	0			
BATC	REWP0902		0.14	554	1	1	7	1	0	0	0	0			
		18	2.58	9980	18	18	126	18	0	0	0	0			

You invoke each report (39 through 41) by specifying:

PRINT REPORT XX

where XX is the report numbers 39 through 41, as control cards in the DBTDBART program.

The sample report(s), column headings, and field descriptions for reports 39 through 41 are identical to those of the DBTRPT38 report.

## DBTFIGHT - Utility and Report

If you need to monitor performance, especially in the realm of contention and record locking, Report 31 is useful, but requires several educated guesses. This report, DBTFIGHT, though not as polished as Report 31, provides more exact information. In fact, in terms of performance, DBTFIGHT may be the most useful report within Tracer.

### To create the DBTFIGHT report

- 1 Run member RUNFIGHT (PGM=DBTRFMT). You must specify these options:  
SEL-TYPE-RUN-UNIT-SELECTION O=ONE, R=RANGE, B=BOTH, A=ANY OF THESE

You must code one or both of these parameters:

```
SEL-RUN-UNIT-NUMBER-1  NNNNNNNNNN  USED WITH 'O', 'R', 'B'
SEL-RUN-UNIT-NUMBER-2  NNNNNNNNNN  USED WITH 'R' or 'B'
```

This forces collection of extra statistics to the //DBTLOCKS file.

- 2 Run member RUNFIGHT (PGM=DBTFIGHT). This reads the //DBTLOCKS file, doing several internal sorts. Its report is done with COBOL DISPLAY statements and shows in the ddname of //SYSOUT.

Parameters are specified in this JCL:

```
EXEC PGM=DBTFIGHT, PARM=' nnn, A, B'
```

where:

*nnn* is the three-digit minimum wait time (in seconds) to include on the report.

*A* is a Y/N value for debug-mode.

*B* is a Y/N value for duplicate PGM combinations. When you specify Y, to reduce the volume of output, this report displays only the worst case problem (where the same two programs are contending).

[Figure 49 on page 154](#) shows a sample of the DBTFIGHT report.

Figure 49 • Sample DBTFIGHT Report

```

DBTFIGHT PGM:  PARM-HOW-LONG-WAIT->100      PARM-YN-DEBUG-MODE->N
              SORT FIELDS=(0001,0004,BI,A,0005,0004,BI,A,0009,0004,BI,D,0013,0004,BI,*
              D,0017,0004,CH,A,0021,0004,BI,A)
-----*
*-B-
DONE WITH INITIAL SORT/REFORMAT:
FROM DBTLOCKS FILE, TO DBTLOCKW FILE
WS-DBTLOCKW-RECS-WRITTEN = 0000048716
-----*
*-E-
              SORT FIELDS=(0025,0002,BI,A,0029,0004,BI,A,0033,0006,PD,A)
WS-DBTLOCKW-RECS-READ = 0000159126
WS-DBTLOCKW-RECS-WRITTEN = 0000000444
              SORT FIELDS=(0153,0004,CH,A,0149,0004,BI,D)
-----*
              --
              THE VALUES AFTER THE H: ARE FOR THE RUN-UNIT THAT HELD THE LOCK,
              --
              THE VALUES AFTER THE W: ARE FOR THE RUN-UNIT THAT WAITED FOR THE LOCK.
              --
-----*

-B- - COMPETING RUN-UNITS FIELDS -
HUND-SECS-INACTIVE:  H: 0000000000  W: 0000008454
INACTIVE-TIME (HR:MN:SC:HS) H: 00:00:00:00  W: 00:01:24:54
JRNL-SEQUENCE-NBR:  H: 0124185504  W: 0124186196
RUN-UNIT-NBR:       H: 0003658292  W: 0003658316
PROGRAM:             H: OCFG01      W: OCFG01
VERB-NBR:           H: 0000000035  W: 0000000035
VERB-NAME:          H: MODIFY       W: MODIFY
RECORD-ID-NBR:      H: 0000000008  W: 0000000008
DP-DATA-OR-PNTRS-ONLY: H: D        W: D
ENVIRONMENT:        H: CICS         W: CICS
PAGE-DBKEY:         H: 0006256771  W: 0006256771
LINE-DBKEY:         H: 0000000011  W: 0000000011
PAGE-GROUP:         H: 0000000000  W: 0000000000
UPDATE TIME IN HHMMSSHHHT FORM -MILITARY HOURS- ( E.G., -1701309015- :
MEANS 1 MINUTE AFTER 5:00 PM AND 30 SECONDS, 90 HUNDRETHS, 15 THOUSANDTHS
)
CLOSE-UPDATE-TIME:  H: 1311211601  W: 1311252983
BGIN-SEQUENCE:      H: 0124180250  W: 0124181828
BGIN-DATE:          H: 0000010120  W: 0000010120
BGIN-TIME:          H: 0013093157  W: 0013100075
END-SEQUENCE:       H: 0124185845  W: 0124186641
END-DATE:           H: 0000010120  W: 0000010120
END-TIME:           H: 0013112439  W: 0013112888
HOW-ENDED:          H: ENDJ        W: ENDJ
TASK-NBR:           H: 0000002997  W: 0000003069
DML-VERB-NBR:       H: 0000000040  W: 0000000029
WHYNOT-CODE:        H: 0000000199  W: 0000000199
TARGET-PAGE-DBKEY:  H: 0006256771  W: 0006256621
TARGET-LINE-DBKEY:  H: 0000000011  W: 0000000002
TARGET-PAGE-GROUP:  H: 0000000000  W: 0000000000
TARGET-REC-ID#:     H: 0000000008  W: 0000000008
TARGET-SEQ:         H: 0124185504  W: 0124186194
PREV-REC-SEQ-RUN-UNIT: H: 0124185502  W: 0124186190
1ST 16 BYTES OF THE TARGET UPDATED RECORD AS DISPLAYABLE AS POSSIBLE:
H: ^.....
W: 4...
HEX VALUE OF FIRST 16 BYTES OF TARGET DATA AREA:
H: 5F77BF070000000000A002100064040
W: F400001740404040404040404040404040
-E- -----
...
... lines omitted
...
WS-DBTLOCKN-RECS-READ = 0000000444
WS-SORT2-RECS-RTRND = 0000048716
NUMBER OF WAITS FOR IDMS PAGES/DBKEYS DETECTED: 0000000222
WS-CNTNS-SKIPPED-SAME-COMBOS ->0000000096
-----*

WS-NBR-PAGE-SAME-DIFF-RUS ->0000005733
WS-NBR-WAIT-NOT-LONG-ENUF ->0000004494
WS-NBR-NOT-CNCRNT-RUNS ->0000001008
WS-NBR-SAME-DBKEY ->0000000077
WS-NBR-SKIP-NO-PG-SPC-CHNGD ->0000000009
WS-NBR-SKIP-NPSC-MODIFY ->0000000009

**=====**
PROGRAM DBTFIGHT FINISHING
**=====**

```

## Tracer Summary - Record Update Counts and Totals

The Tracer Summary Report contains summary counts of the number of records that were updated both by individual programs and by types of update verbs. The counts are broken down by record-type, and optionally by user ID, and also based on parameter settings.

### Input Values

This section describes the input values for the Tracer Summary Report.

#### YN-CALL-TCRSUMRY

Specifying Y calls the TCRSUMRY sub-program. This program will produce the //TCRSUMRY file which shows summary counts showing the number of times each processed record-type was updated.

When this value is set to N, the TCRSUMRY sub-program is not called; the //TCRSUMRY file is not opened or produced.

#### YN-TCRSUMRY-BY-PROGRAM

When this value is set to Y (Default), a breakdown sub-total line will display showing the number of times each record-type is updated based on the program name which bound the run-unit in which the updates occurred. The program name displayed is the value that is in the PROGRAM-NAME field in the SUBSCHEMA-CTRL block when the application program issues the BIND RUN-UNIT DML statement.

#### Note:

\_\_\_\_\_

This parameter set to Y will nullify the effect of setting the YN-TCRSUMRY-ENVIRONMENT to Y.

\_\_\_\_\_

When this value is set to N, none of the sub-totals at the program level are printed. This is useful in cases where you do not want to see a breakdown by which the program issued the updates or by which type of program environment, batch vs. online.

#### TCRSUMRY-HI-LEVEL-BREAK-CODE

These are the available values:

- |      |   |  |
|------|---|--|
| FILE | = | When this is set to FILE, the sub-totals are based on the DDNAME associated with the FILE where the records that were updated were Stored. |
| HOUR | = | When this is set to HOUR, the sub-totals are based on each hour portion of the time of day when the updates occurred.                      |
| MINS | = | When this is set to MINS, the sub-totals are based on each minute portion of the time of day when the updates occurred.                    |

- SECS = When this is set to SECS, the sub-totals are based on each second portion of the time of day when the updates occurred.
- NONE = When this is set to NONE, there is no high-level breakdown. The counts show only the totals for the entire journal period, rather than being broken down by the other possible settings.
- USER = When this is set to USER, the sub-totals are based on each USERID; counts of updates are shown for each USERID which requested the updates.

### **YN-TCRSUMRY-ENVIRONMENT**

When this value is set to Y, the program Environment type is specified in the field where the program-name would normally be shown, rather than breaking down the counts based on which program issued the update. The Environment type is the four digit code that IDMS uses internally.

These are examples of some of the IDMS environment types:

- BATC = CV-Batch program (usually run with a //SYSCTL DDCARD in the JCL.
- DBDC = Internal run-units where the application program is loaded in the IDMS address space.
- CICS = External run-units where the application program was loaded in a CICS address space.

**Note:** \_\_\_\_\_

Some sites are set up so that if there are multiple CICS address spaces which may have IDMS programs issuing updates to a single IDMS-CV, the values may be different based on the CICS initiating region (e.g., CIC1, CIC2, CIC3, etc.).

---

When this value is set to N (Default), the program-name will be the break value and will be seen on the report.

**Note:** \_\_\_\_\_

If the parameter YN-TCRSUMRY-BY-PROGRAM is set to N, the breakdown for either the program or for environment would be eliminated from the //TCRSUMRY file.

---

### **YN-TCRSUMRY-EXCL-TOTALS**

When this value is set to Y, there will be no sub-total lines displaying on the //TCRSUMRY file. This parameter is available if you wish to use SAS or some other analysis reporting programs to work with the detail data only. In this case these programs will probably produce their own subtotal counts.

When this value is set to N (Default), sub-totals will be produced at each break-point.

#### YN-TCRSUMRY-SPACER-DETAIL

When this value is set to Y (Default), blank (spacer) lines are inserted in the //TCRSUMRY file to make the report easier to view.

When this value is set to N, no spacer lines are inserted in the //TCRSUMRY file. This parameter is provided primarily for cases when the intent in producing the //TCRSUMRY file is to use it as an input file for another program.

### Output Description

The ASG-Tracer Summary Report option allows the user to request a summary report of the number of record updates captured on a particular journal archive. This can be useful when determining the volume of updates on the archive. This information can be of particular interest for ASG-Replication Agent users in order to better determine the volume of record updates that will be issued for the receiving SQL database environment. This information can also be useful for performance purposes in general.

The Summary Report can be displayed by FILE, TIME, USER and program name. The run time environment such as BATCH or ONLINE can also be displayed based on the input criteria. The display of program name and the run time environment is mutually exclusive. Sub-totals and final totals are also produced. The input criterion determines the sub-total breaks.

This section describes the line items found in the Summary Report based on the input parameter used:

#### Input Parameter:

YN-CALL-TCRSUMRY	Y
TCRSUMRY-HI-LEVEL-BREAK-CODE	NONE (Report breaks are by record name only)

#### Report Line:

N:ALL	R:JOB	P:ALL	V:CO #:000000003	S:JOB-EMPOSITION
-------	-------	-------	------------------	------------------

#### Input Parameter:

YN-CALL-TCRSUMRY	Y
TCRSUMRY-HI-LEVEL-BREAK-CODE	FILE

#### Report Line:

D:FILE01	R:JOB	P:ALL	V:CO #:000000003	S:JOB-EMPOSITION
----------	-------	-------	------------------	------------------

#### Input Parameter:

YN-CALL-TCRSUMRY	Y
TCRSUMRY-HI-LEVEL-BREAK-CODE	SECS

(SECS will display the full time HHMMSS, MINS will display only hour and minutes as M:HHMM, and HOUR will display only the hour as H:HH.)

**Report Line:**

S:144947 R:JOB P:ALL V:CO #:000000003 S:JOB-EMPOSITION

**Input Parameter:**

YN-CALL-TCRSMRY Y  
TCRSMRY-HI-LEVEL-BREAK-CODE USER

**Report Line:**

U:USER01 R:JOB P:ALL V:CO #:000000003 S:JOB-EMPOSITION

**Input Parameter:**

YN-CALL-TCRSMRY Y  
TCRSMRY-HI-LEVEL-BREAK-CODE SECS  
YN-TCRSMRY-BY-PROGRAM Y

**Report Line:**

S:144947 R:JOB P:PGM01 V:CO #:000000003 S:JOB-EMPOSITION

**Input Parameter:**

YN-CALL-TCRSMRY Y  
TCRSMRY-HI-LEVEL-BREAK-CODE SECS  
YN-TCRSMRY-BY-PROGRAM Y  
YN-TCRSMRY-ENVIRONMENT Y

**Report Line:**

S:144947 R:JOB E:BATC V:CO #:000000003 S:JOB-EMPOSITION

These are the item descriptions by position:

Position	Description
----------	-------------

- Position 1: N: = NONE Report breaks are by record name only.  
D: = (DD name) File Name  
S: = Time record was updated to the second - HHMMSS  
M: = Time record was updated to the minute – HHMM  
H = Time record was updated to the minute – HH  
U: = User ID
- Position 2: R: = Record Name
- Position 3 P: = Program Name  
E: = DC Run Time Environment
-

Position	Description
Position 4:	V: = DML Verb
Position 5:	#: = Number of records updated
Position 6:	S: = Set Name affected

---



---

## Appendix A

---

### Reformatted File Record Layout

Tracer produces a reformatted IDMS journal archive output file. Your site may use this file to produce reports tailored to your specific requirements. You may use a variety of report languages that are capable of reading sequential files (e.g., Guest-IDT from ASG). The record layout of the reformatted file produced by Tracer simplifies the task of developing audit trail reports.

For example, each record that is written to the reformatted file and that corresponds to a MODIFY PAYROLL-RECORD database update issued by an applications program contains two images of the PAYROLL-RECORD. One image shows the record as it existed before the change (BFOR) and the other image shows the record as it existed following the change (AFTR). When Tracer User Exit 14 is installed, each reformatted record also contains the PROGRAM-NAME, USER-ID, TERMINAL-ID, and many other fields which may be included in an audit trail report itemizing each database change.

### Reformatted File Record Occurrence Layout

This chart documents the layout of each record occurrence in the Tracer reformatted file. This is especially useful if you are using TSO Browse to look at the record occurrences or defining the record layout for a programming language that requires the beginning offset of each field.

Begin Offset	Field Length	Field Type	Field Name
1	4	ALPHA	RFJ-TYPE
5	4	BINARY	RFJ-SEQ
9	4	BINARY	RFJ-RUN-UNIT-ID
13	2	BINARY	RFJ-RFMT-RECORD-LENGTH

Begin Offset	Field Length	Field Type	Field Name
15	2	BINARY	RFJ-PAGE-GROUP
17	1	ALPHA	RFJ-NA-LAYOUT-TYPE-THIS-FILE
18	1	ALPHA	RFJ-YN-THIS-REC-TRUNCATED
19	1	BINARY	RFJ-VERB-NUM
20	1	BINARY	RFJ-DBK-LINE-NBR-BITS
21	4	BINARY	RFJ-DB-KEY
25	2	BINARY	RFJ-USER-REC-ID
27	1	ALPHA	RFJ-HOW-COMPRESSED
28	1	ALPHA	RFJ-REC-FIXED-OR-VARIABLE
29	2	BINARY	RFJ-IMAGE-LEN
31	2	BINARY	RFJ-PREFIX-LEN
33	1	ALPHA	RFJ-YN-IS-REC-COMPRESSED-ON-DB
34	1	ALPHA	RFJ-YN-IS-REC-DECOMPRESSD-HERE
35	1	ALPHA	RFJ-ON-OLD-OR-NEW-IDMS-RELEASE
36	1	ALPHA	RFJ-DATE-NORMAL-CENTURY-RECVRY N = Normal (original date/time form) C = Includes century R = Date/time of end of recovery-unit. In job @TE0BREF when parameter SEL-YN-INCLUDE-CENTURY-N-DATE is set to Y, RFJ-DATE-NORMAL-CENTURY-RECVRY will be C in //DBTRFMTO FILE when you use the parameter to PGM=DBTROLBK. Use ENDING-TIME-RECOVERY-UNIT= Y. RFJ-DATE-NORMAL-CENTURY-RECVRY will be set to R.
37	4	BINARY	RFJ-TASK-ID-NUMBER
37	4	BINARY	RFJ-UNIQUE-RECOVERY-UNIT (REDEF)

Begin Offset	Field Length	Field Type	Field Name
41	4	BINARY	RFJ-VIASET-OWNER-DBKEY
45	4	BINARY	RFJ-VIASET-NEXT-DBKEY
49	2	BINARY	RFJ-RECORDS-KEYLENGTH
51	2	BINARY	RFJ-NBR-RECS-PRC-EARLIER-RUNU
53	1	ALPHA	RFJ-STORAGE-TYPE
WHEN RFJ-DATE-NORMAL-CENTURY-RECVRY = N:			
54	8	ALPHA	RFJ-RUN-UNIT-START-DATE (MM/DD/YYYY)
WHEN RFJ-DATE-NORMAL-CENTURY-RECVRY = C:			
54	2	NUM-D	RFJ-START-DATE-WC-CENTURY
56	2	NUM-D	RFJ-START-DATE-WC-YEAR
58	2	NUM-D	RFJ-START-DATE-WC-MONTH
60	2	NUM-D	RFJ-START-DATE-WC-DAY
WHEN RFJ-DATE-NORMAL-CENTURY-RECVRY = N			
62	8	NUM-D	RFJ-RUN-UNIT-START-TIME-NBR (HHMMSSHS)
WHEN RFJ-DATE-NORMAL-CENTURY-RECVRY = C			
62	2	NUM-D	RFJ-RUN-UNIT-START-HOUR
64	2	NUM-D	RFJ-RUN-UNIT-START-MINUTE
66	2	NUM-D	RFJ-RUN-UNIT-START-SEC
68	2	NUM-D	RFJ-RUN-UNIT-START-HUNDR-SEC
WHEN RFJ-DATE-NORMAL-CENTURY-RECVRY = R and CA-IDMS Release 12, 14, or 15:			
54	5	COMP-3	RFJ-DATE-CENT-YEAR-MONTH-DAY
59	7	COMP-3	RFJ-TIME-HR-MIN-SEC-MILLIONS
66	2	ALPHA	RFJ-DAY-OF-WEEK

Begin Offset	Field Length	Field Type	Field Name
68	2	COMP-3	RFJ-JULIAN-DAYS-INTO-YEAR
WHEN RFJ-DATE-NORMAL-CENTURY-RECVRY = R and CA-IDMS Release 10.2:			
54	2	ALPHA	RFJ-I102-EDT-DATE-CENTURY
56	2	ALPHA	RFJ-I102-EDT-DATE-YEAR
58	2	ALPHA	RFJ-I102-EDT-DATE-MONTH
60	2	ALPHA	RFJ-I102-EDT-DATE-DAY
62	2	ALPHA	RFJ-I102-EDT-DATE-HOURS
64	2	ALPHA	RFJ-I102-EDT-DATE-MINUTES
66	2	ALPHA	RFJ-I102-EDT-DATE-SECONDS
68	2	ALPHA	RFJ-I102-EDT-DATE-HUNDR-SECS
70	8	ALPHA	RFJ-RUN-UNIT-PROGRAM-NAME
78	16	ALPHA	RFJ-RECORD-NAME
94	16	ALPHA	RFJ-VIASET-NAME
94	16	ALPHA	RFJ-CHANGED-SET-NAME (REDEFINED)
110	16	ALPHA	RFJ-VIASET-OWNER-RECORD-NAME
110	16	ALPHA	RFJ-CHANGED-SET-OWNER-NAME (REDEF)
126	8	ALPHA	RFJ-SCHEMA-NAME
134	1	ALPHA	RFJ-WHICH-POINTERS-SET-HAS
135	2	BINARY	RFJ-SCHEMA-VERSION-NUMBER
137	1	ALPHA	RFJ-YN-MODIFY-CHANGED-OR-NOT
138	4	ALPHA	RFJ-BEFORE-IMAGE-PRESENT-IND
142	4	ALPHA	RFJ-AFTER-IMAGE-PRESENT-IND
146	1	ALPHA	RFJ-DATA-OR-POINTERS-ONLY

---

Begin Offset	Field Length	Field Type	Field Name
147	8	ALPHA	RFJ-VERB-IN-CHARACTERS
155	4	ALPHA	RFJ-ENVIRONMENT
159	1	ALPHA	RFJ-YN-THIS-TARGET-OF-DML-VERB
160	8	ALPHA	RFJ-TASK-NAME
168	8	ALPHA	RFJ-TERMINAL
176	2	BINARY	RFJ-USERID-LEN
178	8	ALPHA	RFJ-USERID
186	8	ALPHA	RFJ-USERID-FLD-PROGRAM
186	8	ALPHA	RFJ-DECOMPRESSION-NAME (REDEFINED)
194	8	BINARY	RFJ-R12X-FULL-JRNL-SEQUENCE
202	8	ALPHA	RFJ-DB-SEGMENT-IDMS-RLSE-12
210	8	ALPHA	RFJ-SUBSCHEMA
218	1	ALPHA	RFJ-OM-REC-ROLE-IN-CHNGD-SET
219	1	ALPHA	RFJ-SORT-ORDER-OF-CHNGD-SET
220	1	ALPHA	RFJ-WHO-TO-KEY-ON-AT-INSERT
221	2	ALPHA	RFJ-IC-INDEX-OR-CHAINED-SET
223	2	ALPHA	RFJ-MEMBERSHIP-TYPE-IN-C-SET
225	2	BINARY	RFJ-NBR-DML-VERB-THIS-RUNU
227	2	BINARY	RFJ-1ST-DIFF-OFFSET
229	1	BINARY	RFJ-SET-PNTR-OFFSET-NEXT
230	1	BINARY	RFJ-SET-PNTR-OFFSET-PRIOR
229	2	BINARY	RFJ-B4-REPEAT-END-BYTES-TRNCTD (REDEF) This field is active only when RFJ-YN-THIS-REC-TRUNCATED = Y.

Begin Offset	Field Length	Field Type	Field Name
231	2	BINARY	RFJ-NBR-NON-SYS-RECS-PRCSD
233	4	BINARY	RFJ-PAGE-NUMBER-MATRIX
237	2	BINARY	RFJ-LINE-INDEX-MATRIX
239	8	ALPHA	RFJ-DDNAME-WHERE-REC-STORED
247	10	NUM-D	RFJ-CLOSEST-UPDATE-TIME (HHMMSSHHTT)
257	1	BINARY	RFJ-SET-PNTR-OFFSET-OWNER
258	1	ALPHA	RFJ-SPACEHOLDER
257	2	BINARY	RFJ-AF-REPEAT-END-BYTES-TRNCTD (REDEF) This field is active only when RFJ-YN-THIS-REC-TRUNCATED = Y.
259	1	BINARY	RFJ-AREA-TYPE
260	1	ALPHA	RFJ-YN-DESTRUCTIVE-TRUNCATE
When parameter SEL-DOWNWARD-COMPAT-FILE-FORM = Y is used:			
259	2	BINARY	RFJ-RFMT-REC-LENGTH-OLD-LOC
285	2	BINARY	RFJ-INDEX-BLOCK-COUNT
261	1	BINARY	RFJ-ROLLOVER-SEQUENCE-NBR
262	1	ALPHA	RFJ-THIS-DBKEY-4-SR3-RECORD-CD
263	1	ALPHA	RFJ-NEXT-PRIOR-OWN-ESTAB-CRNCY
264	1	ALPHA	RFJ-NPO-TYPE-OF-PNTR-CHANGED
265	4	BINARY	RFJ-REESTABLISH-CRNCY-DBKEY
269	4	BINARY	RFJ-NEXT-POINTER-CHANGED
273	4	BINARY	RFJ-PRIOR-POINTER-CHANGED
277	4	BINARY	RFJ-OWNER-POINTER-CHANGED
281	4	BINARY	RFJ-ALT-IMAGE-NON-TARG-PTR-CHG

Begin Offset	Field Length	Field Type	Field Name
When parameter SEL-CV-NBR-OR-INDEX-BLK-SZ = I is used:			
285	2	BINARY	RFJ-INDEX-BLOCK-COUNT
When parameter SEL-CV-NBR-OR-INDEX-BLK-SZ = C is used:			
285	2	BINARY	RFJ-CV-NUMBER-FROM-JOURNAL
287	1	ALPHA	RFJ-AUTOMATIC-DML-VERB
288	1	ALPHA	RFJ-B4-OR-AF-NPO-PTRS
288	1	ALPHA	RFJ-TRUNCATED-TRAILING-BYTE This field is active only when RFJ-YN-THIS-REC-TRUNCATED = Y.
289	2	BINARY	RFJ-LENGTH-OF-BFOR-PTRS-IMAGE
291	2	BINARY	RFJ-LENGTH-OF-AFTR-PTRS-IMAGE
293	2	BINARY	RFJ-LENGTH-OF-BFOR-DATA-IMAGE
295	2	BINARY	RFJ-LENGTH-OF-AFTR-DATA-IMAGE
297	100	BINARY	RFJ-BFOR-POINTERS-AREA
397	100	BINARY	RFJ-AFTR-POINTERS-AREA
497	300	IN-DB	RFJ-BFOR-DATA-AREA
797	300	IN-DB	RFJ-AFTR-DATA-AREA

The beginning offset and length values for these two fields assume a length of 100 for their corresponding DEPENDING ON fields:

- RFJ-LENGTH-OF-BFOR-PTRS-IMAGE = 100
- RFJ-LENGTH-OF-AFTR-PTRS-IMAGE = 100

The beginning offset and length values for these two fields assume a length of 300 for their corresponding DEPENDING ON fields:

- RFJ-LENGTH-OF-BFOR-DATA-IMAGE = 300
- RFJ-LENGTH-OF-AFTR-DATA-IMAGE = 300

## DBTRFMTJ File Fields (SEL-YN-DBT-WRITE-REFORMATTED=Y)

This section describes each field that Tracer writes to the DBTRFMTJ file when the SEL-YN-DBT-WRITE-REFORMATTED field is set to Y. The record layout has been provided as a COBOL copybook in the Tracer SRCLIB from the installation tape. There, you will find the COBOL picture and usage clause for all fields in the reformatted file.

These first few fields contain values that were copied directly from the source journal record occurrences. Below these (in the record layout) are fields that Tracer gathers from the load module specified on the DBTRECIN file. Also included are both the BFOR and AFTR images of pointers and data journaled for this updated record occurrence (where the data portion displays as it did to the application program immediately before and after the update verb was issued).

### Source Journal Fields

#### **RFJ-TYPE**

This field contains the type of journal record used when this record occurrence was completely assembled. These are the valid values for this field:

BFOR  
AFTR

#### **SEQ**

This is the journal sequence number of the source journal record when this record occurrence was completely assembled. If this field is a double word on the journal archive, the left-most columns are truncated.

#### **RFJ-RUN-UNIT-ID**

This is the run unit number associated with the run unit used by the updating application program.

#### **RFJ-RFJ-RFMT-RECORD-LENGTH**

COBOL programs use this field to obtain the total length of the current record occurrence. When Cobol reads the record, it does not automatically set a variable with the length of the full record, because this is a variable length record occurrence. Each record occurrence has four fields that are used for lengths. There are four fields that have variable lengths where those lengths depend on the values in their OCCURS DEPENDING ON LENGTH (ODO) fields. These fields are located at the end of each record. The RFJ-RFMT-RECORD-LENGTH field is the sum of all the variable lengthened fields of this record occurrence, plus the length (300 bytes) of the fixed portion of the record-occurrence (located before the ODO fields at the end).

#### **RFJ-PAGE-GROUP**

This field contains the page group associated with the run unit that issued the update.

**RFJ-NA-LAYOUT-TYPE-THIS-FILE**

This field contains a code character that contains either the letter N (Normal) or the letter A (Alternate). You will use the Alternate (A) only in special cases requiring the involvement of ASG technical support personnel.

**RFJ-YN-THIS-REC-TRUNCATED**

(Default setting is N.) This field would be set to Y, which indicates that the record's data portion is truncated. This is true only if you requested truncation of records greater than 10,000 bytes (based on selection criteria). This feature is very seldom needed, but is especially helpful in cases where a record type is compressed on the IDMS database and usually contains large numbers of blanks or low-value characters at the end of the record.

**RFJ-VERB-NUM**

This is the number of the DML verb the application program issued after which IDMS did the update. These are defined in the *CA-IDMS Programmer's Quick Reference Guide*.

**RFJ-DBK-LINE-NBR-BITS**

This is the number of bits in the right-most positions of the dbkey word that are used to specify the line index portion of the dbkey. This setting is based on how the matrix point was specified in the PAGE CONTAINS (#) RECORDS clause in the schema.

**RFJ-DB-KEY**

This is the dbkey of this updated record. There are two fields. One contains only the page number portion of the dbkey, and the other contains only the line index portion of the dbkey. Tracer has split these apart based on how the matrix point was specified in the PAGE CONTAINS (#) RECORDS clause in the schema.

**RFJ-USER-REC-ID**

This is the Record ID number specified for this record type in the schema.

**RFJ-HOW-COMPRESSED**

This is set to a meaningful value only when the type of record that was journaled is compressed on the database (i.e., the RFJ-YN-IS-REC-COMPRESS-ON=DB=Y). These are the valid values:

- C = IDMSCOMP is used to compress the record
- P = PRESSPACK is used to compress the record
- H = A homegrown database procedure is used to compress the record, and the name of the decompression database procedure was specified when Tracer was run via the SEL-DECOMPRESSION-DBPROCEDURE parameter setting.

**RFJ-REC-FIXED-OR-VARIABLE**

F (Fixed) or V (Variable) indicates how the record-type is defined in the schema.

**RFJ-IMAGE-LEN**

This is the length of the journaled database record, including pointers and data (if applicable), and the 44 bytes of control information IDMS places on each journal record.

**RFJ-PREFIX-LEN**

This is the number of bytes in the pointer portion of this record type (the dbkeys stored at the beginning of the record that are used to show its position in all the sets in which it participates).

**RFJ-YN-IS-REC-COMPRESSED-ON-DB**

Y (Yes) or N (No) indicates whether this record type is compressed on the database (i.e., did the schema describe this as a compressed record type?).

**RFJ-YN-IS-REC-DECOMPRESSED-HERE**

Y (Yes) or N (No) indicates whether this record type was decompressed by Tracer and is in this record occurrence in its decompressed form. If the parameter SEL-YN-DECOMPRESS-IF-NEEDED was set to N when you ran Tracer, the values in this record occurrence are still in their compressed form and this field will be set to N.

**Note:** \_\_\_\_\_

When this value is set to N, the RFJ-USERID-PROGRAM-NAME field is set to the name of the DCTTABLE if PRESSPACK that was used, or to the name of the homegrown decompression procedure or IDMSCOMP, depending on the type of compression applicable for this record type.

---

## Load Module Fields (from DBTRACE)

**RFJ-ON-OLD-OR-NEW-IDMS-RELEASE**

This is a code field that contains the letter N (New), if the IDMS software release that was used to produce the journal is any release 10.2 or later or contains the letter O (Old) if the IDMS release is 10.2 or earlier.

**RFJ-DATE-NORMAL-CENTURY-RECVRY**

This field contains a 1 letter indicator code which indicates the format of the fields which occupy the part of the record that initially contains the run-unit starting date and time fields for the updating run-unit.

N = Normal indicates that these layouts are used (shown in [Figure 50 on page 171](#) in COBOL syntax).

**Figure 50 • N=Normal - Layout in COBOL Syntax**

```

05 RFJ-START-DATE-TIME-FIELDS.
   10 RFJ-RUN-UNIT-START-DATE.
       15 RFJ-RUN-UNIT-START-MONTH      PIC 9(02).
       15 RFJ-RUN-UNIT-SLASH-1          PIC X(01).
       15 RFJ-RUN-UNIT-START-DAY        PIC 9(02).
       15 RFJ-RUN-UNIT-SLASH-2          PIC X(01).
       15 RFJ-RUN-UNIT-START-YEAR       PIC 9(02).
   ...
   10 RFJ-RUN-UNIT-START-TIME-NBR        PIC 9(08).
   10 RFJ-RUN-UNIT-START-TIME-GRP REDEFINES
       RFJ-RUN-UNIT-START-TIME-NBR.
       15 RFJ-RUN-UNIT-START-HOUR        PIC 9(02).
       15 RFJ-RUN-UNIT-START-MINUTE      PIC 9(02).
       15 RFJ-RUN-UNIT-START-SECOND      PIC 9(02).
       15 RFJ-RUN-UNIT-START-HUNDR-SEC  PIC 9(02).

```

C = Century indicates that these layouts are used (shown in [Figure 51 on page 171](#) in COBOL syntax).

**Note:**

Use this layout when the following parameter setting is used in the //DBTINPUT parameter file. It is used when Tracer executes the step that reads the journal archive file.

```

SEL-YN-INCLUDE-CENTURY-N-DATE  Y
Y=RFJ-RUN-UNIT-START-DATE=CCYYMMDD

```

**Figure 51 • C=Century - Layout in COBOL Syntax**

```

10 RFJ-START-DATE-WITH-CENTURY REDEFINES
   RFJ-RUN-UNIT-START-DATE.
       15 RFJ-START-DATE-WC-CENTURY      PIC 99.
       15 RFJ-START-DATE-WC-YEAR         PIC 99.
       15 RFJ-START-DATE-WC-MONTH        PIC 99.
       15 RFJ-START-DATE-WC-DAY          PIC 99.

```

R = Recovery-Ending-date-time indicates that the date and time fields of the updated run-unit are associated with the date and time that the recovery-unit ended successfully and not the actual start time. This is shown in [Figure 52 on page 172](#) in COBOL syntax. These fields are placed on the record by the program DBTROLBK when this parameter is used:

```
USE ENDING-TIME-RECOVERY-UNIT      Y
Y=PUT ENDING DATE/TIME IN RFJ-REC
```

**Figure 52 • R = Recovery-Ending-Date-time - Layout in COBOL Syntax**

```
05 RFJ-END-RECOVERY-DATE-TIME REDEFINES
    RFJ-START-DATE-TIME-FIELDS .
    10 RFJ-DATE-CENT-YEAR-MONTH-DAY      PIC 9(8) COMP-3.
    10 RFJ-TIME-HR-MIN-SEC-MILLIONS      PIC 9(12) COMP-3.
    10 RFJ-DAY-OF-WEEK                    PIC X(2) .
    10 RFJ-JULIAN-DAYS-INTO-YEAR          PIC 9(3) COMP-3.
05 RFJ-I102-EDT-END-RECOV-DT-TM REDEFINES
    RFJ-START-DATE-TIME-FIELDS .
    07 RFJ-I102-EDT-DATE.
    10 RFJ-I102-EDT-DATE-CENTURY          PIC 9(2) .
    10 RFJ-I102-EDT-DATE-YEAR             PIC 9(2) .
    10 RFJ-I102-EDT-DATE-MONTH           PIC 9(2) .
    10 RFJ-I102-EDT-DATE-DAY             PIC 9(2) .
    07 RFJ-I102-EDT-TIME.
    10 RFJ-I102-EDT-TIME-HOURS            PIC 9(2) .
    10 RFJ-I102-EDT-TIME-MINUTES          PIC 9(2) .
    10 RFJ-I102-EDT-TIME-SECONDS          PIC 9(2) .
    10 RFJ-I102-EDT-TIME-HUND-SECS       PIC 9(2) .
```

**Note:**

The RFJ-DATE-CENT-YEAR-MONTH-DAY field (for CA-IDMS releases 12.x, 14.x, and 15.x) is produced using the IDMS date stamp and shows the date as *CCYYMMDD*.

**RFJ-TASK-ID-NUMBER**

If this update was done by a run unit associated with an internal online task (DBDC environment), this is the task number of that online task. Otherwise, the value here is unpredictable.

**RFJ-UNIQUE-RECOVERY-UNIT (redefines RFJ-TASK-ID-NUMBER)**

This is a unique number assigned by Tracer for every record occurrence that was updated by a given recovery unit. This can be used to sort by recovery unit. If the selection control parameter SEL-TASK-NBR-OR-UNIQUE-REC is set to U then this redefined field is used.

**RFJ-VIASET-OWNER-DBKEY**

This is set only when these three conditions are met:

- This record is the target of the DML verb and data was stored on this source journal record
- This record type has a VIA storage mode
- This VIA set was defined with OWNER pointers

If the value is less than 3, Tracer was unable to find this value.

**RFJ-VIASET-NEXT-DBKEY**

When this record type is defined as VIA but no owner pointers were defined for the VIA set, this field contains the next dbkey in that set.

**RFJ-RECORDS-KEYLENGTH**

This is the number of bytes from the beginning of the record until the first non-key field in the record. The first non-key field is never a part of the CALC key, never before the CALC key, and never a field that serves as a SORT key for any set in which the record participates.

**RFJ-NBR-RECS-PRC-EARLIER-RUNU**

This field counts the number of records processed by Tracer (based on selection criteria), not including this one so far in this run unit.

**RFJ-STORAGE-TYPE**

This is a code field that contains the letter C (CALC), or V (VIA), or F (Fixed). This tells the technique that IDMS uses to determine how to physically place the record occurrence on the database.

**RFJ-RUN-UNIT-START-DATE**

This is the date on the BGIN journal record for this run unit in the format MM/DD/YY.

**RFJ-RUN-UNIT-START TIME**

This is the time on the BGIN journal record for this run unit in the format *HHMMSSHS*.

where:

*HH* = hours  
*MM* = minutes  
*SS* = seconds  
*HS* = hundredths-of-seconds

**RFJ-RUN-UNIT-PROGRAM-NAME**

This field contains the characters that were in the PROGRAM-NAME field in the subschema control communication block when the BIND RUN UNIT statement was executed by the updating program.

**RFJ-RECORD-NAME**

This is the name of this updated record (as found in the subschema load module in the SR51 record).

**RFJ-VIASET-NAME and RFJ-VIASET-OWNER-RECORD-NAME**

When the record is the target of a DML verb, there is data present, and this record is defined with a storage type of VIA. These fields contain the name of the VIA set and the OWNER RECORD TYPE of that set.

**RFJ-CHANGED-SET-NAME and RFJ-CHANGED-SET-OWNER-NAME (redefines RFJ-VIASSET-NAME and RFJ-VIASSET-OWNER-RECORD-NAME)**

This is the name of the set whose pointers were updated.

This is the name of the record type that serves as OWNER of the set whose pointers were updated.

**RFJ-SCHEMA-VERSION-NUMBER**

The name of the schema and the schema version number associated with the area found in the DMCL tables where this record was stored.

**RFJ-WHICH-POINTER-SET-HAS**

This code indicates which pointers the set has. This was originally specified by the person who built the schema syntax. These are the valid settings for this field:

- A = (ALL) Next, prior, and owner pointers
- N = (NEXT) Only next pointers
- P = (PRIOR) Next and prior pointers
- Q = Owner and next pointers
- Z = Next, prior, and owner pointers on SR8 records

**RFJ-YN-MODIFY-CHANGED-OR-NOT**

This field indicates whether any bytes in the data portion of this record were actually changed (Y) or whether the application program issued a MODIFY verb but did not change any fields in the record (N).

**RFJ-BFOR-IMAGE-PRESENT-IND**

This field indicates the type of before image displayed. These are the valid values:

- BFOR This record contains the before image of the data and the pointers for this database record.
- BFPT This record contains only the before pointers for this record.
- (Blank) This record contains neither before pointers nor data for this record.

**RFJ-AFTR-IMAGE-PRESENT-IND**

This field indicates the type of after image displayed. These are the valid values:

- AFTR This record contains the after image of the data and the pointers for this database record.
- AFPT This record contains only the after pointers for this record.
- (Blank) This record contains neither after pointers nor data for this record.

**RFJ-DATA-OR-POINTERS-ONLY**

This value is D if both the data and the pointers were journaled, or P if only the pointers were journaled.

**RFJ-VERB-IN-CHARACTERS**

These are the valid values:

ERASE  
MODIF  
STORE  
CONNE  
DISCO  
SEL  
ERASE SELECTIVE  
PRM  
ERASE PERMANENT  
ALL  
ERASE ALL

**RFJ-ENVIRONMENT**

This field identifies the IDMS run-unit environment. These are the valid values:

DBDC INTERNAL-CV type of run unit (e.g., ADSO)  
BATC BATCH-CV type of run unit (e.g., using SYSCTL DDCARD)  
CICS Run unit opened by an application program loaded in a CICS address space.

**RFJ-YN-THIS-TARGET-OF-DML-VERB**

A Y in this field indicates that this reformatted record represents a database record that was a direct target of a modification by a DML verb. This reformatted record type is created by the Tracer selection criteria defaults.

An N in this field identifies this reformatted record as one of the database records associated with the target, including pointers, which were updated by a DML verb. This can be useful for Tracer jobs where the DBA is gathering performance information. In order to create reformatted records of this type you need to set the select criteria SEL-YN-ONLY-TARGET-OF-DML-RC to N and be sure that the SEL-YN-RETURN-POINTERS is set to Y, which is the default.

***Additional Filled Fields (Optional - Exit 14 installations only)***

**RFJ-TASK-NAME**

For DBDC environment run units, this is the task code name.

**RFJ-TERMINAL**

For DBDC environment run units, this is the VTAM node name when it is accessible based on the IDMS SYSGEN parameters.

**RFJ-USERID-LEN**

This is the length of the user ID.

**RFJ-USERID**

For DBDC environment run units, this is the user ID of the signon session that initiated this updating run unit.

**RFJ-USER-ID-FLD-PROGRAM**

If the update is a native mode CICS program, this field contains the User ID of the CICS signon session.

**RFJ-DECOMPRESSION-NAME (redefines RFJ-USER-ID-FLD-PROGRAM)**

This field would be used only if records were written to the file in their compressed form. This situation requires the involvement of ASG technical support personnel.

**RFJ-R12X-FULL-JRNL-SEQUENCE**

This field contains the full journal sequence number that is used in CA-IDMS releases 12 and later. It is a full doubleword binary field.

**RFJ-DB-SEGMENT-IDMS-RLSE-12**

For journals produced by CA-IDMS releases 12 or later, this field contains the SEGMENT name associated with the Page-Number and Page-Group associated with the updated record.

**RFJ-SUBSCHEMA**

This is the name of the subschema used by the updating run unit.

**RFJ-OM-REC-ROLE-IN-CHNGD-SET**

This code indicates whether the updated record occurrence participates as an owner or a member.

**RFJ-SORT-ORDER-OF-CHNGD-SET**

This code indicates how the set is ordered. These are the valid settings for this field

S = Sorted  
N = Next  
P = Prior  
F = First  
L = Last

**RFJ-WHO-TO-KEY-ON-AT-INSERT**

This code could be used by a site-written program if that program is trying to store data back into the database records, which were earlier erased by a program containing a bug, or that simply should not have been run.

Often, before a STORE verb can be issued for a record occurrence, the application program must establish proper currencies in the following instances sets:

- Where the record participates as a member
- Where IDMS automatically does the CONNECT when the record is stored.

This code tells the program if it needs to establish currency on the owner record of the RFJ-CHANGED-SET-NAME or on another member record type.

These are the valid settings for this field:

- O = Owner
- M = Member

#### **RFJ-IC-INDEX-OR-CHAINED-SET**

This code indicates the type of set that was updated. These are the valid values for this field:

- CH = Chained set (traditional sets)
- IS = Indexed set system owned (integrated index with SR7 owner)
- IU = Indexed set user owned (integrated index with regular owner)

#### **RFJ-MEMBERSHIP-TYPE-IN-C-SET**

This is a two-digit code that indicates how the records in this set are to be disconnected and connected, based on the settings specified by the person who built the schema. A common setting is MA (Mandatory Automatic), meaning that member record occurrences in this set may only be disconnected through an ERASE verb. When record occurrences of the member record type in the set are stored, IDMS automatically CONNECTs them at the appropriate logical position in the set based on currency and/or the value in the field(s) serving as a sort key (if any) at the time of the STORE.

These are the valid settings:

- MA = Mandatory Automatic
- OM = Optional Manual
- OA = Optional Automatic
- MM = Mandatory Manual

**RFJ-NBR-DML-VERB-THIS-RUNU**

This field counts the number of different DML verbs that were issued by the updating application program so far in this run unit.

If the update verb is one of the ERASE verbs, IDMS may delete more than a single non-system type of record occurrence during one DML verb. This number increases once per each individual record occurrence that is deleted.

**RFJ-1ST-DIFF-OFFSET**

When the updating verb is a MODIFY, this field contains the offset where the first difference is found. This occurs when comparing the data image of the updated record before and after the MODIFY verb was executed.

**RFJ-SET-PNTR-OFFSET-NEXT**

This is a one-byte binary field that tells the offset into the prefix portion of the record where the next pointer for the changed set is located.

**RFJ-SET-PNTR-OFFSET-PRIOR**

This is a one-byte binary field that tells the offset into the prefix portion of the record where the prior pointer for the changed set is located. If the set does not have PRIOR pointers, then this field contains the value 1 (hexadecimal 01).

**RFJ-B4-REPEAT-END-BYTES-TRNCTD**

Use this field only when the user-specified parameter settings allow a very large IDMS record to be truncated. In rare cases, this field contains the length of the identical recurring bytes at the end of the BEFORE DATA fields that were truncated. (See also ["RFJ-YN-THIS-REC-TRUNCATED" on page 169.](#))

**RFJ-NBR-NON-SYS-RECS-PRCSD**

This field counts the number of normal records that were not automatically maintained as system-type records updated during this run unit so far.

**RFJ-PAGE-NUMBER-MATRIX**

This is the Page-Number portion from the dbkey of the record that was updated. This was extracted from the dbkey using the appropriate number of bits from the dbkey specified in the schema. This is based on how you initially formatted the IDMS dbkey. The normal (default) setting for IDMS sites indicates that the first 3 bytes of the dbkey fullword (24 bits) are set aside to contain the Page Number portion of the dbkey. This field contains only the Page-Number.

**RFJ-LINE-INDEX-MATRIX**

This is the Line-Index portion from the dbkey of the record that was updated. This was extracted from the dbkey using the appropriate number of bits from the dbkey which was specified in the schema. It is based on how you formatted the IDMS dbkey. The normal (default) setting for IDMS sites indicates that the last 1 byte (8 bits) of the dbkey fullword are set aside for containing the Line Index portion of the dbkey. This field contains only the Line-Index number.

**RFJ-DDNAME-WHERE-REC-STORED**

This is the name of the DDNAME that is specified in the DMCL where the updated record is located when the Journal is produced.

**RFJ-CLOSEST-UPDATE-TIME**

This is the time that is taken from the closest TIME record on the Journal. It shows approximately the time that the application program issued the update DML verb. Usually, the time indicated in this field is not as useful as the time that Recovery-Unit was completed, which is the time that IDMS removed the update Lock and thus when other IDMS users could see the update. See [Figure 52 on page 172](#) for information about the Recovery-Ending-Date-time Layout in COBOL Syntax.

RFJ-END-RECOVERY-DATE-TIME is populated when the program DBTROLBK runs (if you specify the selection parameter value USE ENDING-TIME-RECOVERY-UNIT.)

**RFJ-SET-PNTR-OFFSET-OWNER**

This is a one-byte binary field that tells the offset into the prefix portion of the record where the owner pointer for the changed set is located. If the set does not have owner pointers, this field contains the value 1 (hexadecimal 01).

**RFJ-AF-REPEAT-END-BYTES-TRNCTD**

Use this field only when the user-specified parameter settings allow a very large IDMS record to be truncated. When this occurs, this field contains the length of the identical recurring bytes at the end of the AFTER DATA fields that were truncated. (See also ["RFJ-YN-THIS-REC-TRUNCATED" on page 169.](#))

**RFJ-AREA-TYPE**

This field contains the code where this record resides. These are the valid values:

- 0 = Native IDMS
- 1 = Native VSAM KSDS
- 3 = Native VSAM RRDS
- 4 = IDMS extent area

**RFJ-YN-DESTRUCTIVE-TRUNCATE**

(Default is N.) Set this field to Y if you requested that Tracer truncate non-reoccurring data located at the end of very large IDMS record occurrences (those where the data portion is larger than 10,000 bytes).

**RFJ-RFMT-REC-LENGTH-OLD-LOC**

This field contains the same value as described in ["RFJ-RFJ-RFMT-RECORD-LENGTH" on page 168.](#) It occurs twice in the record occurrence for compatibility with earlier versions of Tracer.

**RFJ-INDEX-BLOCK-COUNT**

This is the number of slots that are kept in the integrated system records (SR8) for the changed set. This is true when the changed set is an integrated index type of set.

**RFJ-ROLLOVER-SEQUENCE-NBR**

This field is almost always the value zero. However if the IDMS journal sequence number is so large that it no longer fits into the doubleword binary journal-sequence-number field, then this field is set to non-zero. This indicates that a rollover condition had occurred.

**RFJ-THIS-DBKEY-4-SR3-RECORD-CD**

This is a code field. This field may contain these valid values: N, Y, or 2.

The value N is most commonly used. N indicates that the record that was updated was not physically stored on the database as a relocated record.

The value Y indicates that an SR3 record type was on the journal for this update because this record occurrence was a relocated record on the IDMS database. This means Tracer was not using the //VSAMIN dataset as a way to determine the real dbkey associated with this updated record occurrence. When this value is Y, the dbkey is the one for the SR3 record occurrence.

The value 2 indicates that this update is for a relocated record occurrence and that the dbkey in this record is the one associated with the SR2 record. Tracer used the //VSAMIN file (produced earlier in Setup steps) in the process of determining the real DBKEY for this relocated record.

**RFJ-NEXT-PRIOR-OWN-ESTAB-CRNCY**

This code indicates the type of pointer from the changed set that Tracer placed into the RFJ-REESTABLISH-CRNCY-DBKEY field. These are the valid settings for this field:

- N = Next pointer type
- P = Prior pointer type
- O = Owner pointer type

**RFJ-NPO-TYPE-OF-PNTR-CHANGED**

This indicates the type of pointer that was changed in the set. These are the valid settings for this field:

- N = Next pointer type
- P = Prior pointer type
- O = Owner pointer type

**RFJ-REESTABLISH-CRNCY-DBKEY**

The dbkey that Tracer extracts from the pointer position of the updated record and places in this field to make it easier for a site written program to do a logical rollback operation. This is the dbkey that the program will probably use to reestablish currency in a set prior to issuing a STORE verb to restore a record that was earlier erased.

**RFJ-NEXT-POINTER-CHANGED**

This is the dbkey that was in the position of the prefix set aside to contain the next pointer for the set where one of the pointers was changed.

**RFJ-PRIOR-POINTER-CHANGED**

This is the dbkey that was in the position of the prefix set aside to contain the prior pointer for the set where one of the pointers was changed. If there are no prior pointers defined for the changed set in this changed record type, then this field is set to the value of 1.

**RFJ-OWNER-POINTER-CHANGED**

This is the dbkey that was in the position of the prefix set aside to contain the owner pointer for the set where one of the pointers was changed. If the record being processed is an owner of the changed set, and there are owner pointers, this field is set to 2. However, if the record being processed is a member of the changed set and there are no owner pointers, this field is set to 1.

**RFJ-ALT-IMAGE-NON-TARG-PTR-CHG**

This is the dbkey that was in the after image if the other pointers were from the before image (or vice-versa) for the type of pointer indicated by the RFJ-NPO-TYPE-OF-PNTR-CHANGED field.

**RFJ-CV-NUMBER-FROM-JOURNAL**

This is the number of the IDMS-CV (specified in the SYSGEN) that is associated with the IDMS-CV job which produced the input journal archive file.

**RFJ-AUTOMATIC-VERB**

This indicates whether this updated record's DML pointers are being adjusted by IDMS to cause an automatic DISCONNECT or CONNECT operation. For instance, when an ERASE DML verb is issued, IDMS automatically DISCONNECTs the record from all the sets in which it participates as a member of the ERASE operation. All of the record occurrences whose pointers are changed in order to do this are said to be operated on by an automatic DISCONNECT verb and hence this field contains the character D. These are the available settings for this field:

D = DISCONNECT

C = CONNECT

**RFJ-B4-OR-AF-NPO-PTRS**

This indicates whether the next, prior, and owner pointer fields were extracted by Tracer from the before (B) or the after (A) image of the journaled record occurrence. These are the available settings for this field:

D = The pointers are from the before image

A = The pointers are from the after image

**RFJ-TRUNCATED-TRAILING-BYTE**

In instances where Tracer needed to truncate a very large IDMS record occurrence, and many of the last bytes in the record consisted of the same character occurring repeatedly (for instance, many blanks together), this field contains the recurring character that was truncated (for instance a blank or a low-value character).

**Note:**

See ["RFJ-YN-THIS-REC-TRUNCATED" on page 169](#) and ["RFJ-B4-REPEAT-END-BYTES-TRNCTD" on page 178](#).

The next four fields (names starting with RFJ-LENGTH) serve as the DEPENDING ON fields used in OCCURS DEPENDING ON situations for the four fields that follow them.

**RFJ-LENGTH-OF-BFOR-PTRS-IMAGE**

If this record occurrence contains the before image of all the pointers for this record, this is the length of those pointers.

**RFJ-LENGTH-OF-AFTR-PTRS-IMAGE**

If this record occurrence contains the after image of all the pointers for this record, this is the length of those pointers.

**RFJ-LENGTH-OF-BFOR-DATA-IMAGE**

If this record occurrence contains the before image of the data for this record, this is the length of that data.

**RFJ-LENGTH-OF-AFTR-DATA-IMAGE**

If this record occurrence contains the after image of the data for this record, this is the length of that data.

**RFJ-LENGTH-OF-BFOR-PTRS-IMAGE**

If this record occurrence contains the before image of all the pointers for this record, this is the length of those pointers.

**RFJ-LENGTH-OF-AFTR-PTRS-IMAGE**

If this record occurrence contains the after image of all the pointers for this record, then this is the length of those pointers.

**RFJ-LENGTH-OF-BFOR-DATA-IMAGE**

If this record occurrence contains the before image of the data for this record, then this is the length of that data.

**RFJ-LENGTH-OF-AFTR-DATA-IMAGE**

If this record occurrence contains the after image of the data for this record, then this is the length of that data.

**RFJ-BFOR-POINTERS-AREA**

This field always starts at the same location in the record if its contents are present at all (see RFJ-LENGTH-OF-BFOR-PTRS-IMAGE). The length of this field is the value in RFJ-LENGTH-OF-BFOR-PTRS-IMAGE and it contains all the Set pointers (dbkeys in the prefix area) as they displayed on the database prior to this update.

**RFJ-AFTR-POINTERS-AREA**

The location where this field starts in this record is dependent on the number in the RFJ-LENGTH-OF-BFOR-PTRS-IMAGE field. If it is zero, then this field starts just after the RFJ-LENGTH-OF-AFTR-DATA-IMAGE field and occupies the number of bytes found in the RFJ-LENGTH-OF-AFTR-PTRS-IMAGE field. If the RFJ-LENGTH-OF-BFOR-PTRS-IMAGE field is not zero, then this field starts just after the RFJ-BFOR-POINTERS-AREA field. This field contains all the set pointers (dbkeys in the prefix area) as they displayed on the database after this update.

**RFJ-BFOR-DATA-AREA**

The location where this field starts in this record is dependent on the numbers in both the RFJ-LENGTH-OF-BFOR-PTRS-IMAGE field, and the RFJ-LENGTH-OF-AFTR-PTRS-IMAGE field. If they are both zero, then this field starts just after the RFJ-LENGTH-OF-AFTR-DATA-IMAGE field and occupies the number of bytes found in the RFJ-LENGTH-OF-BFOR-DATA-IMAGE field. This is where the DATA portion of the updated record is placed, as it existed prior to this update verb. The length of this area is the value in the RFJ-LENGTH-OF-BFOR-DATA-IMAGE field (which is the maximum length of this record type, as defined in the schema).

**RFJ-AFTR-DATA-AREA**

The location where this field starts in this record is dependent on the numbers in the RFJ-LENGTH-OF-BFOR-PTRS-IMAGE field, the RFJ-LENGTH-OF-AFTR-PTRS-IMAGE field, and the RFJ-LENGTH-OF-BFOR-DATA-IMAGE fields. The DATA portion of the updated record is placed in this field, as it existed after this update verb. The length of this area is the value in the RFJ-LENGTH-OF-AFTR-DATA-IMAGE field (which is the maximum length of this record type, as defined in the schema).



---

## Appendix B

---

### Parameters Affecting Relocated Records

If you have restructured some record types on a database, there is a chance that IDMS may have relocated some record occurrences. In most instances, Tracer is able to recognize this situation and use the journaled SR3 records to determine which record type was restructured. However, if you find odd-looking data in your reports, it might be that Tracer failed to recognize some restructured record type. This appendix contains a brief explanation of the restructure/relocation scenario and parameters that are supplied in case Tracer fails to automatically detect restructured/relocated records.

Relocation occurs if there is insufficient space on the database page at the time of the restructure that increases the specified size of the record occurrence. In this situation, relocated record occurrences on the database and on the IDMS journal archive files are represented as at least two distinct physical database records. The stub record points at the relocated record (SR2 and SR3 records, respectively).

Tracer can reverse the relocate function return the record images to the original format that was shown to the updating application program. For this to happen, you must provide Tracer with information about any records that you have restructured and that may still have relocated record occurrences on your databases.

When provided with pointer-length, area name, whether the record is compressed, and whether it is of variable length, Tracer has enough information to determine which record type an SR3 record really represents.

This information is required because the IDMS software does not always journal the SR2 record type, which indicates the schema-defined Record ID number associated with the journaled SR3 record.

The fields listed below enable you to obtain this type of information. You may describe up to ten different restructured record types.

**Note:** \_\_\_\_\_

If you do not need to use all ten restructured record definitions, set the SEL-RESTRUCT-REC-PTR-LN-xx field to 12345 (where xx is 01 to 10) in the definition block following the block being used. If you do not need to use any definition blocks, set the first definition block (where xx is 01 or SEL-RESTRUCT-REC-PRT-LN-01) to 0000012345.

---

The RECDDES report, which is produced by the IDMSRPTS utility, can provide this parameter information for the restructured record selection options:

**SEL-RESTRUCT-COMPRES-D-F-01**

Indicates the type of compression used. Available selections are:

- C = IDMSCOMP
- P = PRESSPACK
- N = Not Compressed
- H = Other

Default = C

You may specify up to ten different parameters (F-01 through F-10).

**SEL-RESTRUCT-REC-AREA-N-01**

Indicates the area name in which the restructured records reside. This field must be 16 bytes; pad with spaces if necessary. This must be the same value as the value in FAC56 DSECT in the subschema.

Default = SAMPLE-AREA-N-01

You may specify up to ten different parameters (N-01 through N-10).

**SEL-RESTRUCT-REC-PRT-LN-01**

Indicates the length of pointers on the restructured records.

Default = 12345

You may specify up to ten different parameters (LN-01 through LN-10).

**SEL-RESTRUCT-RECORD-NBR-01**

Contains the Record ID of the restructured records.

Default = 1020

You may specify up to ten different parameters (NBR-01 through NBR 10).

**SEL-RESTRUCT-VARIABLE-F-01**

Indicates whether the record is variable length. Available selections are:

V = Variable

N = Not Variable

You must specify N if the record is not compressed.

Default = N

You may specify up to ten different parameters (F-01 through F-10).

This list shows the defaults for the restructured record definition block:

SEL-RESTRUCT-REC-PTR-LN-01	0000012345
SEL-RESTRUCT-REC-AREA-N-01	SAMPLE-AREA-NM01
SEL-RESTRUCT-COMPRESD-F-01	C
SEL-RESTRUCT-VARIABLE-F-01	N
SEL-RESTRUCT-RECORD-NBR-01	0000001020



---

## Appendix C

---

# CA-IDMS System Exit 14 Advanced Customization Techniques

This appendix provides advanced customization procedures for the CA-IDMS System Exit 14.

## Customization Procedures

Use the DCMT VARY MEMORY command to enable the debug capabilities by changing the applicable characters in the CWA area.

RHDCUX14 examines these characters to determine if it should write debug messages to the log and whether it should issue Task or System Snaps.

The first time that you invoke CA-IDMS System Exit 14, the value RUNNUX14 is moved to the address at offset 390 (decimal) into the CWA area (UX14 Workarea).

A two-step process is required

### *To determine the UX14 Workarea address in your IDMS CV*

- 1 You must first determine the address of the CWA in your IDMS CV by issuing this command:

```
DCMT D MEM CSA +310
```

The first full word that is displayed in the memory block shown on the screen is the address of the CWA area in your CV.

- 2 Add 390 (hex 186) to the address of the CWA to determine the address of the RHDCUX14 10-byte workarea. Once you have determined this UX14-Workarea address, you can display the value by issuing the command:

```
DCMT D MEM (UX14-Workarea)
```

If you determined the correct address from the two steps above you should see the characters RUNNUX14 at the UX14-Workarea address.

- If you see the value RUNNUX14x0, (last character is a zero value) then the Tracer RHDCUX14 program is indicating that it has not encountered an error condition. (In this case, you may decide you do not need any debug information. However, if you still want debug information, continue these steps).
- If you see the value RUNNUX14D#, (second to last character is D) then RUNNUX14 is indicating that it has encountered a debug logic path at some point during this IDMS CV run. In this case, the # character position contains a non-zero value. This value is the logic path number associated with the error condition that Tracer encountered. A description of each possible logic path number is listed in the table at the end of this chapter. This is an example command to invoke the CA-IDMS System Exit 14 debug mode:

Example:

To invoke the Tracer RHDCUX14 debug mode, first issue this command which tells RHDCUX14 to send two #WTL (Write To Log) commands each time it writes a journal record:

```
DCMT V MEM (UX14-Workarea) C'RAT'
```

The information written to the IDMS journal that follows the Tracer literal contains the logic path number that was taken. The second message contains the actual data that was written to the IDMS journal through the #PUTJRNL command, if it was issued based on the logic path taken. This also notifies RHDCUX14 to issue a task #SNAP, in addition to the #WTL statements, if logic path errors are encountered.

**Note:** \_\_\_\_\_

If you are the only person using the IDMS CV in your testing area, you might want to use the DCMT WRITE STATISTICS command to flush the LOG buffer after you have caused some update run units. This ensures that you are able to see any #WTL messages produced by RHDCUX14 through the online Print Log (OLP) transaction.

You can issue a FIND TR000001 command to locate the #WTL messages and/or a FIND RHDCUX14 command to locate the title line to a #SNAP that was produced by RHDCUX14. If you want to discontinue writing the log messages, you may return to the default settings using this command:

```
DCMT V MEM (UX14-Workarea) C'RUNN'
```

---

To determine the meaning of bytes and associated values, refer to the character position shown in the first column of this table. The second column shows the value to which the DBA may change this position in order to get the indicated type of debug information (described in column three):

R U N N U X 1 4 ? ? (The original value)

0 1 2 3 4 5 6 7 8 9 (Character Position #)

POS. NO.	CHG. POS.	Debug Feature Put Into Effect By The Change
0		Character position 0 shows if RHDCUX14 is running (R) or has been stopped (S). You should not need to change this field.
1		Character position 1 directs RHDCUX14 to begin debug processing. This causes RHDCUX14 to write messages to the log based on the type of logic path taken.  These are the possible values you may enter:
1	E	This directs RHDCUX14 to write messages to the log only when an error logic path is taken (see logic path number descriptions below).
1	A	This directs RHDCUX14 to write messages to the log for all logic paths.  There are two types of TR000001 debug messages written to the log; they come in pairs (i.e., whenever one type is written, the other type is written immediately following the same run unit).  Message type 1 - After the ASGTR: literal is the 1 digit logic path number that was taken (see below).  Message type 2 - After the ASGTR: literal is the user journal record that was either written to the IDMS journal or was in the process of being constructed with the intent of being written later.
2		Character position 2 directs RHDCUX14 to produce a #SNAP dump to the log when it encounters a particular debug logic path.  There are two types of #SNAPS you can tell RHDCUX14 to produce:

POS. NO.	CHG. POS.	Debug Feature Put Into Effect By The Change
2	T	Directs RHDCUX14 to do a TASK type of #SNAP.
2	S	Directs RHDCUX14 to do a SYSTEM type of #SNAP.  When you specify a value here you also specify the logic path number in character position 3. This combination of values directs RHDCUX14 to perform the #SNAP you specify the next time it encounters this logic path number.  <b>Note:</b> _____ After RHDCUX14 has issued the #SNAP, it does not continue to #SNAP each time it hits that logic path (only the first time). It keeps track of this by using character position number 5 as a flag.
3		Character position 3 directs RHDCUX14 to the type of logic path number you want debug information for. This is only used if you want to tell RHDCUX14 to produce a SNAP when RHDCUX14 takes a particular logic path number (the number you put in this position). Therefore, you change this (character position number 3), only when you also change character position number 2 to tell RHDCUX14 to produce a SNAP to the log dump.
	#	If you put a number in this character position, RHDCUX14 only does a #SNAP when it encounters the logic path associated with that number.
	N	If you leave this as is (N), then RHDCUX14 #SNAPs on any logic path number type. You will know which logic path number RHDCUX14 is taking upon an error condition through one (or both) of these ways: <ol style="list-style-type: none"> <li>1) Issue the DCMT D MEM (UX14-Workarea) command after some update run units have been processed in the CV. If there is a non-zero number in character position number 9, that is the problem logic path number.</li> <li>2) By the value displayed in the log at the end of the first TR00001 message pair. For RHDCUX14 to produce this message upon encountering a debug logic path, you must have changed the value in character position number 4.</li> </ol>

POS. NO.	CHG. POS.	Debug Feature Put Into Effect By The Change
4		Character position 4 directs RHDCUX14 to give you debug information even though RHDCUX14 may not determine any problems.
4	G	If you set this character position to G, RHDCUX14 issues a task snap if other parameters request it, even when it uses an acceptable logic path (0). That is, even if Tracer did not detect any problem in getting the information it needed, it still produces the debug information requested (based on the values you specify in the fields described below). If you leave this character at its initial setting (U), there will be no memory #SNAPS to the log when RHDCUX14 determines it has found everything it needs to write a USER journal record.
5		Character position 5 is used by RHDCUX14 as a flag to determine if it has already issued a #SNAP. RHDCUX14 sets this flag to X the first time that you invoke RHDCUX14 during a particular IDMS CV run. When RHDCUX14 issues a #SNAP (because you changed character position number 2), this value is changed to P.
	X	If you want to get more than one #SNAP during a single IDMS CV run (e.g., if you want a #SNAP for more than one type of logic path) and RHDCUX14 already did one #SNAP, you can change the logic path number to the next one you want #SNAPped, and then set this back to its original value of X.
6 7		Character positions 6 and 7 should always contain the number of this exit (14). Do not change these two positions.
8		Character position 8 either contains a space or a D. If it is set to D, RHDCUX14 has encountered a debug logic path condition sometime during this IDMS CV run. Do not change.
9		Character position 9 is set by RHDCUX14. You should not change this value. The one-byte number placed here (in decimal) indicates the last logic path number that occurred.

---

POS. NO.	CHG. POS.	Debug Feature Put Into Effect By The Change
		<p><b>Note:</b></p> <p>This field is initialized to 0. If any non-zero logic path is taken and that number is written here, if RHDCUX14 properly handles another run unit, it does not overwrite the earlier non-zero value. If a non-zero value is in character position 9, a problem logic path was taken earlier. You can obtain more debug information by changing some of the other character position numbers, as documented.</p> <p>Brief explanations of each possible logic path follow.</p>
9	0	This is not an error condition; this indicates that RHDCUX14 has not found any problems. Thus, it wrote a user journal record with the values it expected to find.
9	1	Indicates that RHDCUX14 was passed a non-zero value for the ERE address, but the supposed ERE control block contains a zero where the address of the ESE should be.
9	2	Indicates that RHDCUX14 was passed a non-zero value for the ERE address, but the TCEERUS bit flag implies that this is not an external run unit.
9	3	Indicates that RHDCUX14 was processing an internal run unit but the TCETDEA field is zero; thus, no task definition element.
9	4	Indicates that the #PUTJRNL command did not complete successfully. If this happens, a message is written to the log with the displayable return code number returned from the macro. The CA-IDMS System Exit 14 turns itself off for the rest of the CV run.
9	5	Indicates that the SSCVIBA offset in the subschema communications block address that was passed to RHDCUX14 did not point to a legitimate variable subschema (VB50).
9	6	Indicates that the non-zero LTESONRC address field in the LTE did not point at a legitimate sign-on RCE control block.

### **Problem Resolution Procedure**

If you call ASG Customer Service and they request a dump of the problem, send as many of these as possible. (This will greatly expedite problem resolution.)

- System message log
- JCL listing at run time
- SYSOUT DD messages
- A complete SYSDUMP (not ABEND AID)
- The last three pages of any reports (or copies of reports)



---

## Appendix D

---

### Creating a Dbkey of Relocated Records

For sites that allow relocated records, Tracer automatically identifies the record type of an SR3 record. However, Tracer does *not* automatically provide the dbkey of the SR2 record in the relocated record pair. *Run the jobs in this appendix only if you must have a dbkey of the SR2 record.* Each of these jobs' names begin with @R.

The jobs whose names begin with @R are optional. They are used to gather from your database information about relocated records. These jobs are needed primarily by sites that use Tracer to propagate changes to the IDMS database to a parallel database environment. The @R jobs enable Tracer to obtain the dbkey of the SR2 record when an update to a relocated record is encountered on the journal. The @R jobs are found in the JCLLIB uploaded from the installation tape.

#### Run Job @RAALOOK

This job executes the IDMS utility IDMSLOOK to gather information that is stored in the global DMCL. Tracer will be able to learn about the DSN= names used in dynamic allocations. This information is saved in an output file that Tracer uses as input to the next job @RABJCLB that will automatically build all the JCL steps necessary to sweep the database file for relocated records.

#### Run Job @RABJCLB

This job executes the program TCRJCLBD that builds JCL to run the TRSR2SWE program that sweeps through an IDMS database file looking for SR3 and SR2 records. The job @RABJCLB creates the JCL that eliminates the need to modify and submit the job @RA0BSWE that also runs the program TRSR2SWE.

Information about relocated records is contained in the database files that are defined to the IDMS-CV. Only one database file is processed in one execution of TRSR2SWE and thereby creating the need to create a multi-step job. TCRJCLBD will build JCL for this multi-step job that will execute TRSR2SWE for each physical database file based on the DMCL that is being used. This program will also produce the //SORTIN data to run with the sort utility to concatenate all of the output files that were produced by the job that runs the TRSR2SWE program.

The TRSR2SWE program requires this information about each database file that it reads:

- The page size of the database file
- The page group that the file will have assigned to it
- The number of bits used in the line-index for the DBKEY format
- The DDNAME associated with the file
- The type of file structure (i.e., BDAM or VSAM)

This is an example of the JCL used to run TRSR2SWE and can be generated by TCRJCLBD:

**Figure 53 • JCL Used to Run TRSR2SWE**

```
//STEPNAME EXEC PGM=TRSR2SWE,  
//  PARM=('DDNAME=FILEDD,PAGESIZE=4276,LINEBITS=08',  
//      'PAGEGRP=0001,DBTYPE=BDAM')  
//*-----  
//FILEDD  DD DSN=THEDSNFILENAME,  
//      DISP=SHR  
//*  
//OUTDCB  DD DSN=TCR.HLQ.CV#.SR2SR3.FILEDD,  
//      DCB=(RECFM=FB,LRECL=12,BLKSIZE=27996)  
//      SPACE=(CYL,(10,10),RLSE),  
//      DISP=(,CATLG,KEEP),  
//      UNIT=SYSDA  
//*
```

### **Program Input**

This program requires these input files that provide the information needed to build the JCL:

//DBTINPUT: This file contains the site-specific parameter settings.

//IDMSRPTS: This file is the output from the IDMSRPTS utility produced in the job @TA2BDM1 where the DMCLST parameter was specified.

@TA2BDM1 needs to be run before this job, although it appears later in the JCL stream.

//IDMSLOOK: This file is the output from the IDMSLOOK utility produced in the previous job, @RAALOOK, where the DMCL ALL SORTED parameter was specified.

//CVJCL: This file contains the JCL used to start up the IDMS-CV.

//SIZEFILE: This file allows the user to tell this program the amount of primary and secondary cylinders that should be allocated for the output file, //OUTDCB. This file will hold the relocated SR2/SR3 information. This information allows you to override the storage requirements by database file DDNAME.

//EXCLUDES: This file allows you to identify a list of files, by DDNAME, that will be excluded from the search for relocated records.

### Input parameters for //DBTINPUT

#### **YN-DEBUG-MODE and DEBUG-NBR-TO-DISPLAY**

Used for debugging and trace purposes by ASG technical support.

#### **PRIMARY-CYL-ALLOC and SECONDARY-CYL-ALLOC**

Used to allow overrides for primary and secondary space allocations for the output files generated by TCRJCLBD that will be used TRSR2SWE.

#### **HIGH-LEVEL-QUALIFIERS**

Used to supply a standard high level qualifier for all intermediate datasets.

#### **TCR-PASSWORD-LOADLIB-DSN**

Allows the program to include the dataset that contains the ASG authorization module in the STEPLIB.

#### **TCR-MAIN-LOADLIB-DSN**

Allows the program to include the Tracer 7.3.0 loadlib in the STEPLIB concatenation.

#### **YN-USE-IDMSLOOK-FILE**

Allows the program to bypass the use of the IDMSLOOK report as an input file. Bypassing the IDMSLOOK report can save CPU cycles when a site does not use the dynamic allocation capabilities of a IDMS release. (N = bypass)

#### **YN-USE-TIME-QUALIFIER and YN-USE-DATE-QUALIFIER**

Date and time can be added as nodes into the dataset name for output and intermediate files.

### DISPLAY-ALL-INPUT-FILES

Allows TCRJCLBLD to echo the entire contents of all the input files

Example: (Parameters begin in column 2; parameter values begin in column 29.)

(If an asterisk is entered in column 1, it indicates a comment.)

Figure 54 • DISPLAY-ALL-INPUT-FILES Parameters

YN-DEBUG-MODE	N
DEBUG-NBR-TO-DISPLAY	0
PRIMARY-CYL-ALLOC	003
SECONDARY-CYL-ALLOC	005
HIGH-LEVEL-QUALIFIERS	IDMS.TCR.Rxxx
TCR-PASSWORD-LOADLIB-DSN	IDMS.TCR.Rxxx.PASSWORD.LOADLIB
TCR-MAIN-LOADLIB-DSN	IDMS.TCR.Rxxx.DIST.LOADLIB
YN-USE-IDMSLOOK-FILE	Y
YN-USE-TIME-QUALIFIER	N
YN-USE-DATE-QUALIFIER	N
DISPLAY-ALL-INPUT-FILES	N

### Input for //SIZEFILE

DDNAME PRI,SEC

Example: EMPDEMO 003,002

### Input for //EXCLUDES

DDNAME

Example:

????DB\*

\*LOD

You may use the masking characters, asterisk or question mark. If, for example, you have index areas assigned to files that contain only SR8s and if the naming convention you use allows a way to identify those files, you can specify that naming convention to exclude all of the index only file based on the matching DDNAME.

Example 1: IX\* (Example begins in column 2.) (If an asterisk is entered in column 1, it indicates a comment.)

This will cause any files to be excluded where the first two digits of the DDNAME is IX.

Example 2: \*IX

This will cause any files to be excluded where the last two digits of the DDNAME is IX.

Example 3:     ??IX\*

This will cause any files to be excluded where the third and fourth characters of the DDNAME is IX, without regard to what characters come after the fourth position.

### **Program Output**

This program will produce these output files:

//JCLSWEEP: The JCL for the TRSR2SWE job that executes the multiple steps.

//FORSORT: The concatenation for the //SORTIN DD for the subsequent jobs.

## **Run Job @RA0BSWE**

The @RA0BSWE job sweeps your IDMS database and builds a sequential file containing information about all of its SR2 and SR3 records. These records were built by the Restructure utility. Later, each file for each database is concatenated and sorted. Then, a VSAM file is built that contains all relocated information for all files in a particular CV.

## **Run Job @RB0BSOR**

Job @RB0BSOR sorts all of the SR2SR3 sequential datasets that were created earlier.

## **Run Job @RC0BCTL**

Job @RC0BCTL reads the sorted file that was built by the last job (@RB0BSOR) and produces the VSAM DEFINE control cards. These control cards will be used by the next job (@RD0BVSM) to allocate an appropriately sized VSAM dataset to be accessed at Tracer runtime when SR3 records are encountered on the journal archive tapes.

## Run Job @RD0BVSM

Before running job @RD0BVSM, you should edit the VSAMCNTL member in the .CNTL PDS for this CV#XXX environment. You can strip the JCL off of that member and change the names as you desire.

## Run Job @RE0BLOD

Job @RE0BLOD loads the SR2/SR3 relationship data to a VSAM file.

---

## Appendix E

---

# Frequently Asked Questions

All of the information presented in this appendix is covered in detail elsewhere in this publication.

- Q.** What is the difference between DBTRFMT and DBTRRFMT?
- A.** DBTRFMT runs against the offloaded IDMS journal archive tape and creates a reformatted journal in the ASG layout and optionally creates reports. The DBTRRFMT reads a reformatted journal and creates reports. Also, DBTRFMT prints Report 01, Report 02, and Report 05, while the only numbered reports printed by DBTRRFMT are Report 11 and Report 05.
- Q.** How do I create a report against the reformatted file?
- A.** Run the DBTRRFMT utility @TG0BRRF. Refer to [Chapter 3, "Getting Started with Tracer," on page 19](#), and [Chapter 5, "Control Parameters," on page 37](#), for more information.
- Q.** When must I specify a schema (map) name on my //DBTINPUT?
- A.** Only when asking for the Tracker field-level audit report or when using SQL-like select statements.
- Q.** What is a schema map?
- A.** A schema map is a load module unique to Tracer. It contains IDD information about all the element names and offsets in a schema. (A subschema load module does not contain element names.) Tracer can quickly access this information without needing to access the IDMS data dictionary at run time.

The schema map load module is created by running the jobs described in ["Step 2 - DMCL and Schema Mapping" on page 21](#), the schema mapping utility, against the IDMS data dictionary (IDD).

The load module must be accessible in your //STEPLIB any time you run the Tracker report or use any field-level selection criteria.

- Q.** When would I need to run the DBTROLBK (rollback) utility?
- A.** If you do not run the DBTROLBK utility, your reports could include updates to records by programs that may have later abended and been rolled-back by IDMS. The DBTROLBK utility removes the non-committed records from the reformatted journal and creates a new reformatted journal without those records on it. Even if you run Tracer at night, when your IDMS CV has quiesced, your journal tapes probably contain non-committed record updates from programs that later aborted and rolled-back. If you do not run DBTROLBK, those records will print on your Tracker report, and you will have no idea that they rolled-back. DBTROLBK works closely with the non-quiesced journal archive feature.
- Q.** When would I need to use the non-quiesced journal archive feature?
- A.** When you are running reports immediately as the journals are offloaded, or when you need to make sure that run-units in progress are not included in your reports (since they might still abend, causing a rollback). Basically, when you turn the non-quiesced journal archive feature on, the DBTRFMT writes information about the unfinished run-units to the //DBTUNFIN file.
- Q.** If I do not get any reports and my job has not abended, what do I do?
- A.** Check the statistics printed at the bottom of the //SYSOUT output file. One of your selection criteria may have been in error, causing records to be omitted erroneously. The table shows the number of times each SEL- statement and //FLDSELS SELECT statement was tested. If the number of rejects is the same as the number of times tested, then that means that none of the records on the journal passed your select criteria.

Also, make sure you are using the correct subschema library (i.e., test for test and production for production) pointing to the correct input file.

See [Chapter 5, "Control Parameters," on page 37](#) for an example.

- Q.** Do I have to code any COBOL programs to use field-level selection?
- A.** You can use the SQL-like selection criteria (specified in the //FLDSELS DD input). If you have selection criteria not easily handled by the SQL selects, you may wish to write your own field-level selection program.

**Q.** What if I want to create my own customized reports using the journal data?

**A.** You have two options:

- Write your own COBOL program which reads the user-friendly reformatted journal and creates the report
- Write a COBOL subprogram (similar to a user-exit), which can be called directly from Tracer.

**Note:** \_\_\_\_\_

If you like the Tracker report, but wish to make minor changes to it, Tracer provides the COBOL source code. If you do make changes, this limits your upward compatibility with future releases.

---

**Q.** Why are there so many parameters to Tracer?

**A.** A job generally only needs about 2 or 3 critical parameters to run. Additional parameters allow customization of reports to your liking. By creative use of //DEFAULT files, your Tracer system administrator can set up the preferred defaults for your sight.

The critical parameters for DBTRFMT (utility which reads the offloaded journals) are:

```
SEL-RU-ENVIRONMENTXXXXX (unless you want the default of DBDC)
```

plus any of these parameters for the reports you wish to print:

```
PRINT REPORT 01Y
PRINT REPORT 02Y
PRINT REPORT 05Y
PRINT REPORT 11Y
PRINT REPORT 30Y
ETC . . .
TRKR-YN-PRINT-REPORTY
```

If you are running the Tracker report and/or using SQL-like selection criteria, this parameter is required:

```
TRACKER-SCHEMA-NAMEXXXXXXXXX
```

**Q.** What if I put a DBTRRFMT parameter on a DBTRFMT job or vice-versa?

**A.** This does not present a problem. If it is not used by the program it is ignored. 95% of the parameters are identical so you could copy //DBTINPUT from one job to another. (See "[Selection Criteria](#)" on page 91.)

- Q.** Is there a way to change the order the data appears on reports?
- A.** Produce a reformatted file, sort by your criteria, feed the sorted file as input to DBTRRFMT (@TGOBRRF).
- Q.** When looking at reports, why are journal sequence numbers grouped by two or more?
- A.** IDMS writes out one journal record for the before image and another for the AFTER image. Thus, one update after another for the same run-unit typically begins grouping by two. Tracer combines the before and after image into one record and reports the first journal sequence number. Also, Tracer combines fragmented records that are original separate journal sequence numbers.
- Q.** Why would I have data on Report 02 (DBOL/DMLO updates) but no data on Report 01?
- A.** A parameter can turn off your selection criteria for just Report 01. This allows you to run Tracker for a particular subschema, but in the same run get a report of ALL DMLO/DBOL updates to any subschema.
- Q.** Why not run all reports from the DBTRFMT utility directly against the journal?
- A.** The journal contains updates that may have later been rolled-back. If you do not want these records included on your reports, you must run the DBTROLBK utility and then run the DBTRRFMT utility to create your reports from the reformatted files. See the next question for additional information.
- Q.** Why do BJ, EJ, COMMITs, and ABORTs sometimes appear on Report 01 and 11 and sometimes do not?
- A.** When running directly against the journal, these record types are available. However, these record types are not written to the reformatted file (to save disk space). When you run Report 01 and/or 11 from the reformatted files using the DBTRRFMT utility, these record types do not print.
- Q.** How does Tracer know which DMCL to use, since there is no DMCL parameter?
- A.** Tracer gets the DMCL name from the DBTRECIN file created in ["Step 2 - DMCL and Schema Mapping" on page 21](#).

In CA-IDMS 12.x, 14.x, or 15.x, the DMCL used is the global DMCL.

**Q.** How can I limit my Tracker report to print a record only when a specific field changed?

**A.** Use SQL-like selection criteria, for example:

```
//DBTINPUT DD *
TRKR-YN-USE-FLDSELS          Y
//FLDSELS      DD *
  SELECT EMPOSITION WHERE
      BEFORE SALARY-AMT-0420 NE AFTER SALARY-AMT- 0420
```

**Q.** How can I limit the element names that are printed on the Tracker report?

**A.** Use the field include feature, for example:

```
//DBTINPUT DD *          Y
  TRKR-YN-USE-FLDINCL
//FLDINCL
  EXC ALL
  INC EMPOSITION          SALARY-AMT-0420
  INC EMPOSITION          SALARY-GRADE-0420
```



## Symbols

#PUTJRNL command 194  
#SNAP  
    dump 191  
    task 190  
#WTL  
    messages 190  
    statements 190  
//DBSTAMP file 2  
//DBTRUNUI file 2  
//MEMRYIN files 31  
//MEMRYOUT files 31  
@IA0BALC job 15  
@IB0BPSW job 15  
@RA0BSWE job 201  
@RAALOOK job 197  
@RABJCLB job 197  
@RB0BSOR job 201  
@RC0BCTL job 201  
@RD0BVSM job 202  
@RE0BLOD job 202

## Numerics

80-column mode 8

## A

abend S806 81  
abort file 2  
ABRT 1  
advanced user topics 9  
AFTER image 5, 86  
aggregate functions 79  
applications 3  
audit trail routines 3  
automatic rollback (in-flight) 5

## B

BEFORE image 5, 86  
BGIN record 135  
BIND UNIT RUN verb 143  
BIND/READY exit 2

## C

CA-IDMS 35  
CA-IDMS System Exit 14 2, 9  
Central Version 5  
CHECKPOINT verb 140  
COBOL  
    DBTSELCT program 99  
    programs 5  
COMMIT verb 144  
Comparison operators 76  
comparison operators  
    SQL 80  
Concurrent Run Unit Summary reports 8,  
    30, 131  
Concurrent Run Units Detail report 8, 30,  
    130  
Concurrent Run Units Summary report 145  
Concurrent Run Units with Suspected  
    Record Locks report 8  
CONNECT verb 35  
CONNECT/DISCONNECT verb 6  
control parameters 37  
conventions  
    naming 13  
conventions page viii  
CV 5

## D

database (tuning/statistics) reports 130  
DBA Statistics/tuning functionality 8  
Dbkey  
    creating relocated records 197  
DBTDBART program 148  
DBTFIGHT 8, 131  
DBTINPUT  
    file 72  
DBTRFMT 92  
    program 91  
    utility 206  
DBTROLBK utility 9, 204  
DBTRPT30 8, 30, 130  
DBTRPT31 8, 30, 130

- DBTRPT33 8, 30, 130
- DBTRPT34 8, 30, 130
- DBTRPT37 8, 30, 131
- DBTRPT38 8, 30, 131
- DBTRPT39 8, 30, 131, 151
- DBTRPT40 8, 30, 131
- DBTRPT41 8, 30, 131
- DBTRRFMT 97
  - parameters 91
  - program 91
  - selection criteria 90
- DCMT VARY MEMORY command 189
- DCMT WRITE STATISTICS
  - command 190
- decompress images 5
- default
  - files 48, 72, 205
  - parameter file 72
  - qualifiers 87
- defaults
  - site-defined 72
- defragment images 5
- DELTA qualifiers 88
- despan images 5
- DISCONNECT verb 35
- E**
  - EARLIEST qualifiers 87
  - elements 74
  - ENDJ 1
  - end-of-job information 2
  - ERASE verb 87
  - excluding records/elements 74
  - Exit 14 9, 189
- F**
  - field-level
    - reports 7
    - selection 7, 74
  - fields
    - masking 56
  - free format input 76
  - frequently asked questions 203
  - functions 80
- H**
  - hex
    - parameters 40
- I**
  - I12XUX14 16
  - I14XUX14 16-17
  - IDMS
    - network database 35
    - IDMS utility 197
    - IDMS/R DBMS 5
    - IDMSLOOK
      - IDMS utility 197
    - IN (SQL predicate) 79
    - incremental refresh 3, 35
    - Individual Run Unit Resource report 151
    - input
      - for //EXCLUDES 200
      - for //SIZEFILE 200
  - J**
    - jobs
      - beginning with @R 197
    - journal file 2
    - journal sequence number 206
  - L**
    - language selection criteria 7
    - LATEST qualifiers 87
    - libraries 12, 17
    - Library Allocation Job 15
    - limitations, SQL 85
    - LINES-PER-PAGE 47
    - logical operators 76
  - M**
    - masking fields 56
    - mathematical statements 86
    - MODIFY verb 87
  - N**
    - NAMELIKE function 82
    - naming conventions 13
    - NBR-ABRT-RECS-TO-DISPLAY 38
    - NBR-CKPT-RECS-TO-DISPLAY 38
    - NBR-PRSD-RECS-TO-DISPLAY 38
    - NBR-STMP-RECS-TO-DISPLAY 38
    - NBR-SUSP-RECS-TO-DISPLAY 38
    - NBR-TO-PROCESS 38
    - NBR-TO-REPORT 38
    - non-executable load module 7
    - non-IDMS parallel file 3, 35
    - non-quiesced
      - journal archive 204
      - journal archive tapes 31
      - journals 9
  - P**
    - parallel
      - databases 35
      - IDMS data 35

- parameters
    - control 39
    - defining 38
    - input for //DBTINPUT 199
    - pointer-only processing 42
    - printing data in Hex 40
    - run control 38
    - see Control Parameters 37
    - selection criteria 48
    - site-written reports 46
    - Tracker report 44
  - PARM-A1 through PARM-A10 47
  - PARM-N1 through PARM-N10 47
  - performance analysis (database) 3
  - presspack database procedures 37
  - PRINT REPORT 01 47
  - PRINT REPORT 02 47
  - PRINT REPORT 05 47
  - PRINT REPORT 11 47
  - Product Authorization Job 15
  - Program 198
  - program input 198
  - Program Resource Reports 8, 131
  - Program Resource reports 30, 131
  - program resource reports 30
- Q**
- quotes, single/double 77
- R**
- READY verb 140
  - record elements update report 2
  - record level reports 6
  - Record Locks report 135
  - Record Occurrence Selection Criteria 56
  - records 74
  - recovery 4
  - RECRPT-COLUMNS 40
  - RECRPT-DBKEY-DISPLAY-MODE 40
  - RECRPT-HEX-DISP-FORMAT 41
  - RECRPT-HEX-DISP-MODE 41
  - RECRPT-HEX-LAST-N-BYTES 41
  - RECRPT-HEX-LENGTH 42
  - RECRPT-HEX-SCALE-LINE 42
  - RECRPT-HEX-START-POS 42
  - RECRPT-HEX-STOP-POS 42
  - RECRPT-YN-DISPLAY-ALL-BATCH 47
  - RECRPT-YN-SHOW-CHANGED-BYTES 42
  - RECRPT-YN-STACK 47
  - Reformat
    - facility 6
  - reformatted journal archive file 2
  - reformatted journal records
    - splitting 34
  - Reformatter reports 6
  - Report
    - 01 109
    - 02 114
    - 05 118
    - 11 125
    - 30 131
    - 31 135
    - 33 140
    - 34 141
    - 37 145
    - 38 148
    - 39 151
    - 40 152
    - 41 152
    - field-level 7
    - record-level 6
  - Report Control parameters 39
  - reports 6
  - RHDCUX14 189
  - rollback 5
  - rollforward processing 5
  - run control parameter 38
  - Run Unit reports 8, 30, 130, 141
  - Run Unit Resource report 148
  - Run Unit Selection Criteria 49
  - Run Units Associated with Record Locks
    - reports 30, 130
  - RUNNEX14 189
- S**
- schema navigation program 7
  - SEL-1ST-NBR-OF-RUN-UNITS 60
  - SEL-1ST-NBR-OF-TASK-IDS 60
  - SEL-1ST-NBR-SELECTED-RECORDS 60
  - SEL-2ND-ELEM-LENGTH 59
  - SEL-2ND-ELEM-OFFSET 59
  - SEL-ABEND-SEQUENCE-NBR 57
  - SEL-CHANGED-ELEM-LENGTH 60
  - SEL-CHANGED-ELEM-OFFSET 60
  - SEL-DBKEY-LINE-INDEX 57
  - SEL-DBKEY-PAGE-NBR-END-RANGE 57
  - SEL-DBKEY-PAGE-NUMBER 57
  - SEL-DDNAME-MASK 57
  - SEL-DEBUG-SEQUENCE-NBR 57
  - SEL-DECOMPRESSION-DBPROCEDURE 58
  - SELECT
    - keyword 76
    - statement 76, 85, 204

- selection criteria
  - parameters 48
- selection criteria masking 55
- SELECTOR-SUBPROGRAM-NAME 47
- SEL-ELEM-VALUE-1-REC 60
- SEL-ELEM-VALUE-2ND 59
- SEL-EXEMPT-RECORDNAME 60
- SEL-EXEMPT-SUBSCHEMA-MASK 49
- SEL-FV-FIXED-OR-VAR-OUTREC 61
- SEL-HUND-SECS-WAIT-B4-COMT 61
- SEL-LOCAL-OR-CV-JRNL 61
- SEL-MAX-BYTES-FOR-DATA 62
- SEL-MAX-BYTES-FOR-POINTERS 62
- SEL-NAME-OF-ONLY-CHANGED-SET 62
- SEL-PROGRAM-MASK 49
- SEL-RU-BEGIN-DATE 49
- SEL-RU-BEGIN-TIME 50
- SEL-RU-END-DATE 50
- SEL-RU-END-TIME 51
- SEL-RU-ENVIRONMENT 52
- SEL-RUN-UNIT-NUMBER-1 52
- SEL-RUN-UNIT-NUMBER-2 52
- SEL-SCHEMA-NAME 63
- SEL-SELECT-PAGE-GROUP-NBR 62
- SEL-SINGLE-RECORDNAME 58
- SEL-SUBSCHEMA-MASK 53
- SEL-SUBSCHEMA-NAME 63
- SEL-TASK-ID-NUMBER 53-54
- SEL-TASK-NAME-MASK 54
- SEL-TERMINAL-MASK 54
- SEL-TYPE-RUN-UNIT-SELECTION 54
- SEL-TYPE-TASK-ID-SELECTION 55
- SEL-TYPE-UPDTS-USER-SYS-ALL 63
- SEL-UPDATE-TIME-START 63
- SEL-UPDATE-TIME-STOP 63
- SEL-USE ENDING TIME RECOVERY UNIT 53
- SEL-USER-BLT-RUAL-TYPE-PROG 63
- SEL-USERID-MASK 55
- SEL-VERBNAME-MASK 64
- SEL-YN-ABEND-MODE 64
- SEL-YN-ALWAYS-SEL-DBOL-DMLO 64
- SEL-YN-CHECK-2ND-ELEM 58
- SEL-YN-CHECK-ELEM-CHG-1-REC 58
- SEL-YN-CHECK-ELEM-VALUE-1-REC 58
- SEL-YN-COMBINE-BFOR-AFTR-OUT 64
- SEL-YN-DBKEY-IN-ANY-PNTR-POS 64
- SEL-YN-DBKEY-IN-CHGD-PTR-POS 65
- SEL-YN-DBOL-DMLO-IN-BOTH-FILES 65
- SEL-YN-DBTRUNUI-BGIN-COMT 65
- SEL-YN-DBT-WRITE-REFORMATTED 65
- SEL-YN-DECOMPRESS-IF-NEEDED 66
- SEL-YN-DEFRAGMENT-IF-NEEDED 66
- SEL-YN-DE-RELOCATE-IF-NEEDED 66
- SEL-YN-DESPAN-IF-NEEDED 66
- SEL-YN-GET-SR2-FROM-VSAM 66
- SEL-YN-GLOBAL-SUBSCHEMA 67
- SEL-YN-INCLUDE-CENTURY-N-0 51
- SEL-YN-LIMIT-TO-PAGE-GROUP-C 67
- SEL-YN-ONLY-1-PAGE-GROUP 67
- SEL-YN-ONLY-TARGET-OF-DML-RC 67
- SEL-YN-PROCESS-ENDEVOR-RUS 68
- SEL-YN-PROCESS-RHDCRUAL-RUS 68
- SEL-YN-PROCESS-USER-RUAL-TYP 68
- SEL-YN-RETURN-DMCL-FIELDS 69
- SEL-YN-RETURN-ONLY-CHNGD-MOD 69
- SEL-YN-RETURN-POINTERS 69
- SEL-YN-RETURN-SYSTEM-RECORDS 69
- SEL-YN-SKIP-CONCURRENT-CHKS 55
- SEL-YN-TARGET-PTRO-NOT-CON-DI 68
- SEL-YN-USE-RANGE-FOR-PAGE-NBR 69
- SEL-YN-WRITE-DBOLDMLO-FILE 70
- SEL-YN-WRITE-DBTRUNUI 70
- SEL-YN-WRITE-SYSTEM-RECS-RFMT 70
- SEL-YN-WRITE-USER-DBSTAMP 70
- setup
  - steps 14
- setup process overview 14
- snapshot image 5
- SQL
  - additions 86
  - comparison operators 80
  - implementation limits 85
  - limitations 85
  - predicates 76
  - predicates, non-supported 79
  - use of quotes in 77
  - user functions 80
- SQL-like
  - language 7
  - selection criteria 73
- stamp file 2
- statistics reporting program 6
- STORE verb 87
- SUBSTR (substring) function 80
- summary report 8, 131

summary reports  
     concurrent run units 145  
     resource 148  
     user report by subschema 140  
 SYSGEN 17

**T**

technical requirements 12  
 Tokens 77  
     complex 80  
 Tracer  
     components 6  
     features 2  
     functions 3  
     getting started 19  
     installation 12  
     product files 12  
 Tracker  
     facility 7  
     report 7, 73, 101  
 Tracker report  
     parameters 44  
 TRACKER-SCHEMA-NAME 44  
 TRKR-DBKEY-DISPLAY-MODE 44  
 TRKR-RPT-YN-80-COL-MODE 44  
 TRKR-RPT-YN-HDR-INC-ALWAYS 44  
 TRKR-RPT-YN-INCLUDE-GROUPS 45  
 TRKR-RPT-YN-INCLUDE-NULLS 45  
 TRKR-RPT-YN-INCLUDE-SAME 45  
 TRKR-USERPROG-NAME 45  
 TRKR-YN-CALL-USERPROG 45  
 TRKR-YN-PRINT-REPORT 45  
 TRKR-YN-USE-FLDINCL 46  
 TRKR-YN-USE-FLDSELS 38

**U**

UPDATE locks 144  
 user report  
     by application 8  
 user report by application 130  
 user report by subschema 30, 140  
 user-defined defaults 72  
 USERRPT-SUBPROGRAM-NAME 46  
 user-written journal reports 9  
 user-written SQL functions 80

**W**

WHICH-UTIL-PRINTS-TRKR 46  
 workarea address (UX14) 189

**Y**

YN-APPLY-SELECTOR-TO-RECRPT 47  
 YN-APPLY-SELECTOR-TO-TRKR 46

YN-APPLY-SELECTOR-TO-USERRPT 4  
     6  
 YN-DBKEY-LIMIT-TO-TARGET 47  
 YN-DISPLAY-FLDSELS-TRACE 38  
 YN-DISPLAY-RECRPT-TRACE 48  
 YN-DISPLAY-SELECTOR-TRACE 39  
 YN-DISPLAY-TRACE 39  
 YN-DISPLAY-TRKR-TRACE 46  
 YN-DISPLAY-USERRPT-TRACE 46  
 YN-MIXED-CASE 39  
 YN-PNTRONLY-RECS-TO-RECRPT 43  
 YN-PNTRONLY-RECS-TO-SELECTOR  
     43  
 YN-PNTRONLY-RECS-TO-TRACKER 4  
     4  
 YN-PNTRONLY-RECS-TO-USERRPT 44  
 YN-PRINT-BANNER 48  
 YN-PRINT-FOOTER 48  
 YN-PRINT-NON-TARGET 70  
 YN-READ-DBTUNFIN 39  
 YN-USE-DEFAULT-FILE 39  
 YN-WRITE-DBTUNFIN 39





ASG Worldwide Headquarters Naples Florida USA | [asg.com](http://asg.com)