precisely

Syncsort Optimize DB2
(EZ-DB2)

Tracer & Cache User Guide
Release 9.10

Version 6.16

August 2017

Syncsort DB2 Tracer & Cache User Guide
(formerly EZ-DB2 Tracer & Cache User Guide)

NOTICE

Syncsort Optimize DB2, formerly called EZ-DB2 is designed to be used by licensed users of IBM Db2 for z/OS, as a supplement to this product.

Contents

Figures

Preface

EZ-Tracer® and **EZ-Cache®** are products of Cogito which allow users of **DB2** to monitor DB2 subsystems, capturing the SQL activity into *Trace Workloads* and providing detailed analysis of the SQL activity.

The *Trace Workload* may be analyzed by other EZ-DB2 components such as **EZ-Alerts®** and **EZ-Impact Analyzer®**. It may also be loaded into the EZ-DB2 *SQL Warehouse* by the EZ-DB2 *SQL Warehouse Manager* for further analysis and optimization by the other EZ-DB2 components **EZ-Impact Analyzer®** and **EZ-XOP®**

Related Documentation

- EZ-DB2 Installation Reference
- EZ-DB2 Getting Started and IVP Guide
- EZ-DB2 Tracer & Cache User Guide (this document)
- EZ-DB2 SQL Warehouse User Guide
- EZ-DB2 Reports Guide.
- EZ-DB2 Commands Reference
- EZ-DB2 Alerts User Guide
- EZ-DB2 Impact Analyzer User Guide
- EZ-DB2 Index Analyzer User Guide
- EZ-DB2 Index Optimizer (EZ-XOP) User Guide
- EZ-DB2 Stats User Guide
- EZ-DB2 Housekeeping and Controls User Guide
- EZ-DB2 Tables Reference

THIS PAGE IS INTENTIONALLY LEFT BLANK

New Features

SET PROFILE Command

If you usually use settings which differ from the defaults as standard for most new workloads you may wish to set these as the default initial settings for your new workloads. You can do this by issuing the command **SET PROFILE** when in the command line. This will save the current settings shown on the panel as the default initial settings for your userid and these will be used as the initial settings whenever you create a new workload. See also *Copy Workload Parameters* below.

Refer to the EZ-DB2 Commands Reference for further information.

Copy Workload Parameters

To create a new workload with the same parameters as an existing workload you can now type CREATE *workload-name* in the command line, where *workload-name* is the name of the new workload name to be created.

Refer to the EZ-DB2 Commands Reference for further information.

Automatically Delete Old Traces from Workload

If desired, Tracer will now automatically delete old traces from a workload to free up DASD space.

Refer to Maximum Kept Traces in EZ-Cache Filters and Maximum Kept Traces in EZ-Tracer Filters.

Select IFCIDS to Be Traced

Previously in EZ-Tracer, you could enable/disable the collection of IFCIDs relating to FETCHs. You may now include/exclude other IFCID classes from the trace. Refer to trace filter parameter Trace IFCIDS.

Capture Individual SQL Executions with Host Variable and Literal Values

Although an important feature of EZ-Tracer/Cache is the ability to merge or consolidate all equivalent executions of SQL (e.g. where only the literal values are different), there are situations where it is desirable to see individual Host Variable and Literal values for captured SQL. For example, in the consolidated workload you will see the TOTAL and AVERAGE CPU for all consolidated SQL, and just one representative access path (that would be derived from the 1st occurrence of the consolidated SQL). As some users have pointed out, this does not allow them to see the “outliers”. That is, 99% of the SQL may share the same access path, but a particular host variable or literal values may have an adverse effect leading to degraded performance. It would be desirable to capture the host variable or literal values associated with these “outliers”.

So, the implementation of Host variable or literal value collection in EZ-DB2 has been designed to capture these values for the “outliers only”, identified by high CPU or GETPAGE activity

It is now possible to specify CPU, Elapsed time and GetPage thresholds that trigger the collection of individual SQL instances complete with their associated literal and host variable values, and execution

timestamp. It is also possible to collect individual executions of SQL without literals and host variables, and SQL with negative SQL codes that trigger the threshold values.

New reports are available to display the SQL with host variables literals, and also to generate ad hoc reports to locate SQL instances for example within a particular time span.

Refer to new report option 25.5 – SQL with Host Variables and Literals and Report 25.6 – SQL Executions ad hoc Reports in the EZ-DB2 Reports Guide.

Refer to the Trace Executions in EZ-Tracer trace filters for further information.

Extended XPRINT Command

When displaying either the SQL Text or the SQL Statistics displays you can use the XPRINT command, to print ALL of the associated detail displays for the current SQL, rather than just the current screen.

As before the report is written to the specified SYSOUT class or dataset (and optional member name). This saves the user from having to repeat the XPRINT command on each screen.

Refer to the EZ-DB2 Commands reference for further information.

Display SQL Error Codes

There is a new report (Report 25.4 – SQL with Errors) to display SQL with negative SQL codes.

The report shows the negative SQL code(s) and number of occurrences for each SQL in the workload that had an error. Refer to the EZ-DB2 Reports Guide for further information.

It is also possible to capture the individual executions of SQL with errors even though the SQL may not have exceeded the threshold value that triggers the capture of individual SQL executions.

Refer to SQL with Errors for further information.

Specify AUTHID and QUALIFIER Conversion

Certain applications, typically SAP®, PeopleSoft and JDBC type applications make use of the SET CURRENT SCHEMA as a way of specifying the object qualifiers to be used by an application. This has the benefit of making the application code transportable. However, this presents a problem for EZ-Tracer/Cache as the SET CURRENT SCHEMA may have been issued before the trace started, and hence Tracer/cache has no way of knowing how to properly qualify the subsequent SQL. In the past, we have defaulted to using the current AUTHID as the qualifier; however, this may not be correct. To resolve this issue, we now allow the user to specify the desired mapping of AUTHID to Object Qualifier.

Refer to ERP controls .

Reports Filter Command

New FILTER commands have been implemented on most reports.

To filter an existing report, enter the command FILTER followed by the keyword Program, DBRM, Package, Plan, Collid, Authid, Table or Index as applicable.

e.g., FILTER PROGRAM EC* to display only those report lines where program name start EC as shown above.

Refer to EZ-DB2 Command Reference for further information.

Flag SQL Command

When viewing any report that displays an SQL statement the FLAG SQL command can be used to highlight any SQL statement. This allows you to highlight any SQL statement for further investigation so that when you come back to a report during the same session or later in another session any flagged SQL statement will still be highlighted.

Refer to EZ-DB2 Commands Reference for further information.

Extend Trace Time

Occasionally, a user may wish to extend the duration of a trace that is already in progress.

To extend the trace time enter the command

/F *jobname*,ADD,*nnnn*

Refer to EZ-DB2 Commands Reference for further information.

Display DB2 transformed version of SQL Text

After an Edit & Explain, EZ-Tracer users may now display the SQL text *after transformation* by the DB2 optimizer.

Refer to EZ-DB2 Commands Reference for further information.

Locate Command added for Load Free Form SQL

When loading Free form SQL into a workload, the user enters a dataset name and member name containing the SQL to be added.

The user may also DRILL on the member name field to display a list of the members in the specified dataset.

As there may be many members in the dataset, and to assist the user in finding the required member, the Locate command may now be entered in the command line.

Refer to the EZ-DB2 Commands Reference for further information.

User Controlled Consolidation of Authids

A new user exit has been implemented that allows user manipulation of the trace data before being written to the SQL log file for consolidation processing.

A sample user exit (EZTREXT1) is provided in the installation SRCLIB that modifies the AUTHID based upon a user supplied conversion table.

For example:

AC% is converted to AC_GRP,
DC% is converted to DC_GRP
etc.

In this way, all possible AUTHIDS may be consolidated to a few departmental groups in a way that is meaningful to the user.

Refer to Appendix A.2 – EZTREXT1 for further information.

Total Elapsed Time for Programs

The Program Summary report has been enhanced to include the TOTAL Elapsed time for programs. The report may be sorted on any column including Total elapsed as before.

Refer to the EZ-DB2 Reports Guide for further information.

Add Dynamic Prepare CPU Times to Trace Summary Reports

The Trace Summary Reports by Plan, Collection, Programs, AUTHIDs etc., have been enhanced to show the Dynamic PREPARE CPU times.

Refer to the EZ-DB2 Reports Guide for further information.

Refresh Catalog Statistics

At the end of each trace, statistics are captured to the statistics repository (warehouse) for each new database encountered. These are the statistics that are displayed when a user drills to a database object (e.g. table or index) from the access path display.

The access paths may be generated at the end of the trace, or explicitly using option EZ-Cache option 1.3 or EZ-Tracer option 2.5 to refresh the access paths.

As the statistics were previously captured for a given workload only the first time that the database was encountered, it was possible for the statistics to become *stale*, and possibly not correspond to the statistics that were present at the time the access path was generated.

Therefore, there is now a new option in EZ-Cache and EZ-Tracer to refresh the statistics each time the access paths are generated.

Refer to Access Path Controls and Refresh SQL access paths and statistics.

Recall Workload Datasets

You can specify R against a workload name to issue a RECALL command to DFSMSHSM to recall of the workload datasets. This may save time if the workload has not been viewed for some time.

See Recall datasets.

Set Trace Class and Destination Command

To set the Tracer/Cache defaults for Trace class and Destination.

In the command-line enter:

```
SET CLASS  nn/OFF
SET DEST   OPx/OFF
```

to set the defaults for this and future new traces.

Refer to the EZ-DB2 Commands Guide for further information.

Set Degree ANY for Tracer Explains

It is now possible to specify the Current Degree to be used when explaining SQL in a Tracer/Cache workload.

SQL may be explained in the following circumstances:

- Automatically at the end of the trace process. You may specify that you want to explain ALL SQL or only NEW SQL in the workload.
- You may explain the SQL in the workload explicitly using option 1.3 or 2.5 - Refresh SQL Access Paths and Statistics.
- You may Explain individual SQL using the EE (Edit & Explain Command).

This feature allows the user to perform "*what-if*" analysis to simulate the effect of binding applications with DEGREE ANY.

Refer to Set Current Degree option on the Explain Controls panel.

Report Stage 2 Processing

Report 22 – SQL by Program has a new column to show Stage 2 (for version "E" access paths only).

Access Path drills includes a new column S2 to indicate whether Stage 2 processing took place.

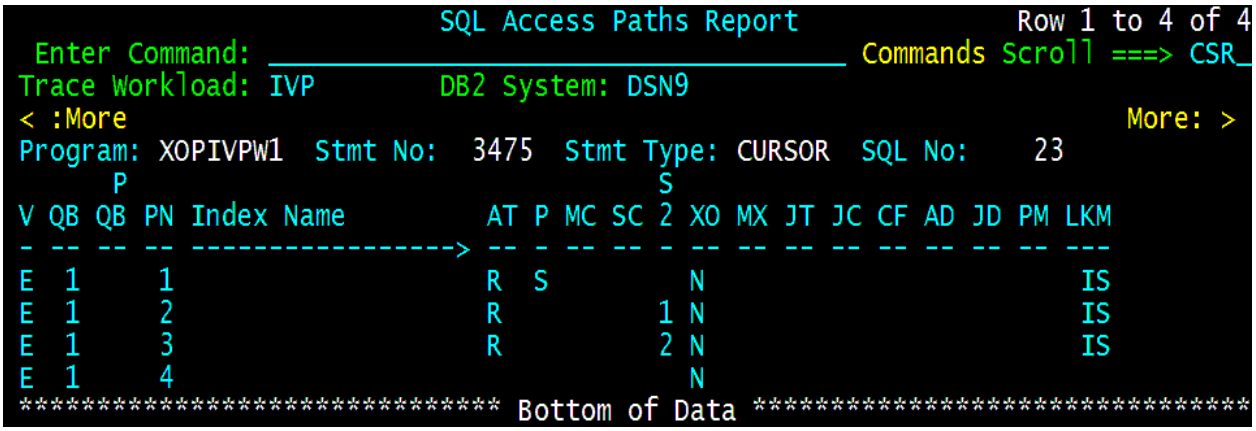


Figure 1 Access Path Stage 2 processing

The column may have the following values:

1 Stage 1 Only
2 Stage 2 processing took place
U DSN_FILTER_TABLE unavailable in which case if DMROWS=RDSROWS then probably no Stage 2 processing.

There is also a new report 25.9 to show all SQL with Stage-2 processing (for Version E access paths only).

For the stage-2 processing to be reported, there must be a valid DSN_FILTER_TABLE created for the plan table used by the explain.

Note: EXPLAIN and BIND plan tables will need to be regenerated to enable the change. Option 9.2.11 can be used to generate a job to create the full plan table set. Optionally edit the DDL to just create DSN_FILTER_TABLE and its Index. The DSN_FILTER_TABLE definition must be V9 or above, i.e. include the MIXOPSEQNO column.

Refer to the EZ-DB2 Reports Guide for further information.

Index Screen Columns

The SQL access path display now includes a new column SC for screen columns.

For ACCESSTYPE I,I1,N or MX this column indicates the number of index keys on top of MATCHCOLS used for index screening.

Refer to the EZ-DB2 Reports Guide for further information.

Report SQL with Errors

There is a new option in EZ-Tracer filters to report ALL SQL with errors when thresholds are active.

See option - Include SQL with Errors. When this option is set to Yes, then all SQL with errors will be included in the SQL with errors reports and Trace execution reports, even if the SQL do not exceed the specified threshold values.

Static SQL in Cache

Prior to DB2 10, EZ-Cache was used exclusively to capture dynamic SQL from the DB2 dynamic statement cache. To capture static SQL, the only option was to use EZ-Tracer, which can capture both dynamic and static SQL.

Some users would use EZ-Tracer to capture both static and dynamic SQL. Others would use a combination of both products, running simultaneous traces with EZ-Cache used for dynamic, and EZ-Tracer with the Start Trace parameter All, Static or Dynamic SQL set to S for static only.

DB2 Version 10 introduces the ability to collect statement level statistics for both dynamic and static SQL statements. Monitor trace class 29 enables the collection of system-wide statement-level statistics for dynamic SQL statements in IFCID 0316, and for static SQL statements in IFCID 0401.

If you are at DB2 10 or higher, you can therefore now use EZ-Cache to capture both dynamic and static SQL. Refer to EZ-Tracer/Cache User Guide Dynamic, Static or ALL Cache for further information.

This means that users now have a near zero overhead method of capturing a representative workload containing both *static* and *dynamic* SQL that can then be used in the other EZ-DB2 components such as EZ-Impact Analyzer, EZ-Index Analyzer and EZ-XOP (Index Optimizer).

Importing CA Detector Data into EZ-DB2

It is now possible to create EZ-DB2 Tracer and Warehouse workloads from SQL captured by the CA Detector® product.

Therefore, it is now possible for users of the CA Detector® product to take advantage of the Index Analyzer and Index Optimizer features of EZ-DB2.

In addition, when viewing the CA Detector data in EZ-Tracer or EZ-SQL Warehouse, CA Detector® users can benefit from the unique SQL consolidation functionality of EZ-DB2.

Refer to CA Detector® documentation for information about externalising Detector® data.

Refer to Appendix Importing CA Detector Data into EZ-DB2 for information about importing the externalized CA Detector® data into an EZ-DB2 workload.

CA Detector® is a program product of CA Technologies Inc.

Trace Buffer Threshold

It is now possible to specify a Trace Buffer Threshold. See the enhanced start trace panel in the following figure.

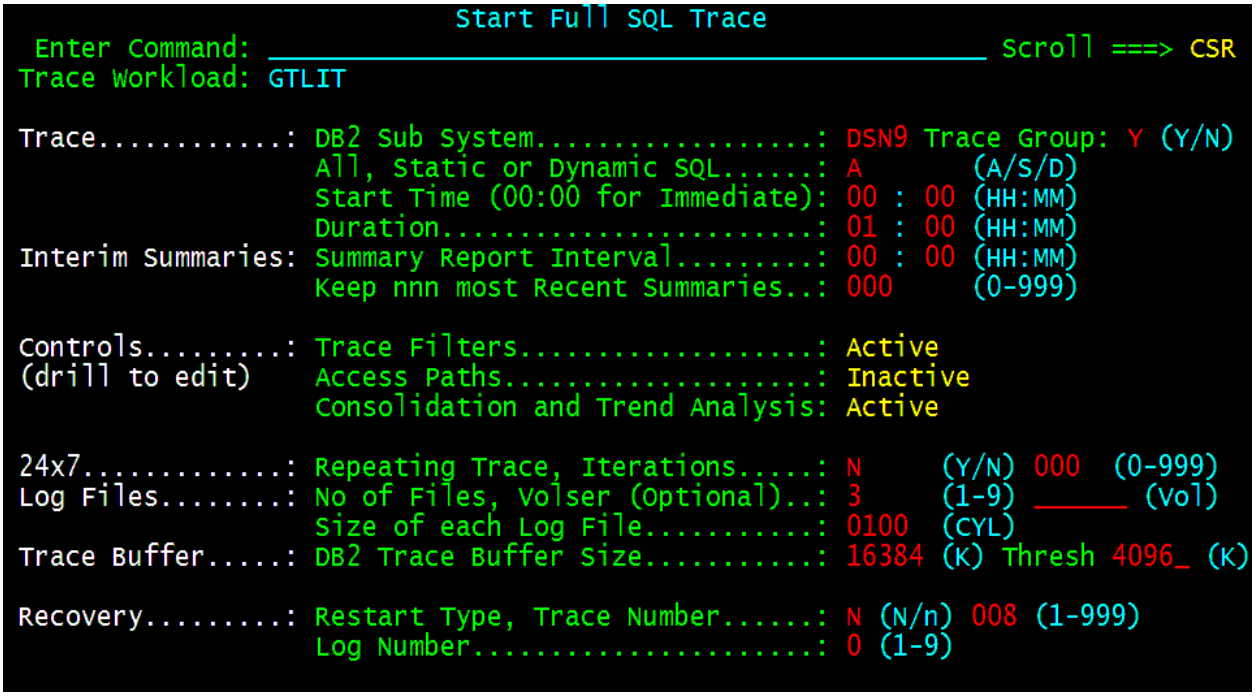


Figure 2 Trace Buffer Threshold

Notice that for the Trace Buffer you can now specify **Threshold** in addition to the **DB2 Trace Buffer Size**.

The Buffer Threshold determines when EZ-Tracer is posted by DB2 to say that there is trace data to be processed. The default threshold is 25% of the Buffer Size, which allows time for EZ-Tracer to process the data before the Buffer fills up.

New Wait Statistics

EZ-Tracer/Cache now captures and displays the DB2 wait statistics and Suspension Timers.

For EZ-Cache these statistics are available in DB2 9 and DB2 10, except where noted some values are available in DB2 10 only.

For EZ-Tracer these statistics are only available in DB2 10.

Note that these statistics are only available if DB2 Accounting Class 3 is active.

The additional statistics captured are as follows:

Tot Wait SYNIO	Accumulated wait time for synchronous IO
Tot Wait Locks	Accumulated wait time for Locks(1)
Tot Wait Switch	Accumulated wait time for Synchronous execution unit switch
Tot Wait Glbl Lock	Accumulated wait time for Global Locks
Tot Wait Agt Read	Accumulated wait time for other agents reads

Tot Wait Agt Write	Accumulated wait time for other agents writes
Tot Wait Latch	Accumulated wait time for latch requests(2)
Tot Wait Pg Latch	Accumulated wait time for page latches(2)
Tot Wait Drain Lck	Accumulated wait time for drain locks(2)
Tot Wait RlseClaim	Accumulated wait time for a drain during wait for claims to be released(2)
Tot Wait Log Write	Accumulated wait time for log writers(2)

(1) Prior to DB2 10 the timer includes latches as well
(2) Available as of DB2 10

Total and Average times are displayed in EZ-Tracer/Cache. Refer to the EZ-DB2 910 Reports Reference for further information.

Add User Identity Field to Tracer Reports

If the User Identity field is added to an SQL, this will now be displayed in the reports for the SQL statement. Refer to the EZ-DB2 910 Reports Reference for further information.

Allow mix of INCLUDE an EXCLUDE Filters

It is now possible to specify both INCLUDE and EXCLUDE filters in EZ-Tracer.

For example, to specify multiple INCLUDE and EXCLUDE filters for AUTHID, DRILL on the AUTHID option in Trace Filters. The panel now includes an eXclude option which may be specified for each parameter,

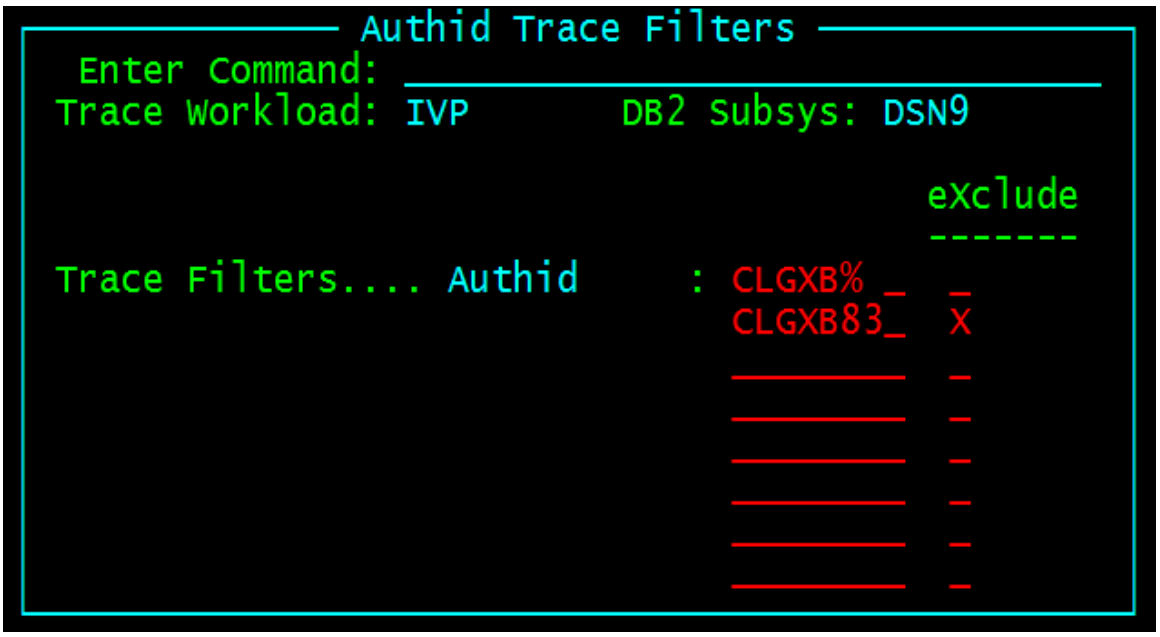


Figure 3 Include/Exclude Filters

Add Collection Name Filter to EZ-Cache

It is now possible to filter a Cache Workload by Collection Name as shown in the following figure.

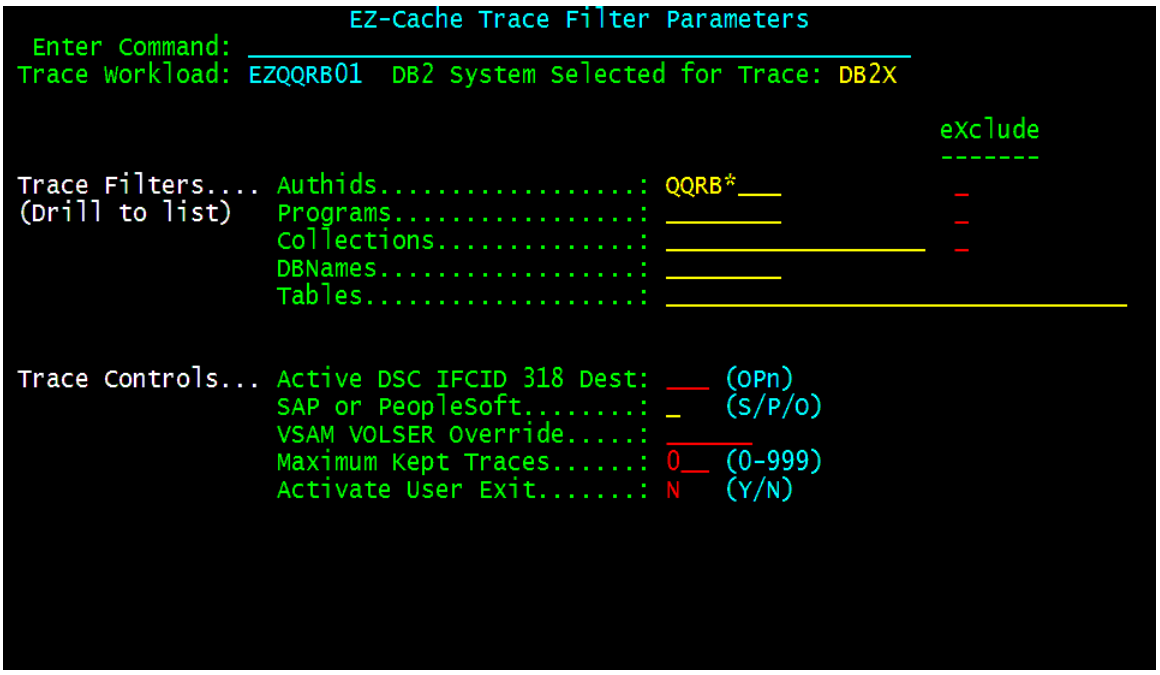


Figure 4 Add Collection Filter to EZ-Cache

You may specify up to 4 Collection names (with wild card support) to be INCLUDED or EXCLUDED from the current workload.

Optional Consolidation of INSERT Statements

The Tracer/Cache Consolidation controls panel has an additional option to enable consolidation of INSERT statements.

INSERTs for the same table will be consolidated when there are no SELECT sub-clauses regardless of how the column list is coded or defaulted.

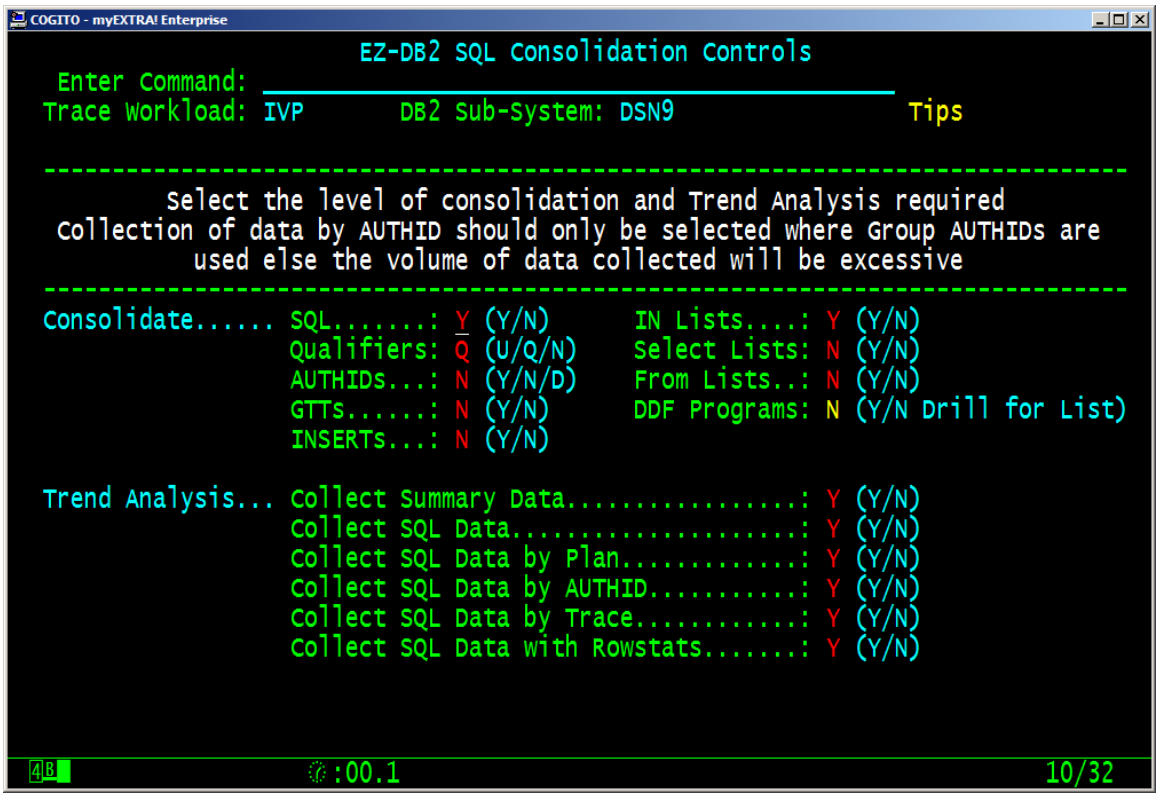


Figure 5 Optional Consolidation of INSERTs

Enter Y to enable consolidation of single row INSERT statements.

INSERTs for the same table will be consolidated when there are no SELECT sub-clauses regardless of how the column list is coded or defaulted.

Allow Start Time > 24:00

To permit users to submit traces with a start time on the following or subsequent day, it's now possible to specify a start time > 24:00.

For example, Start Time = 32:00 to start Trace at 8:00 am tomorrow.

Running Tracer and Cache as Started Task

It is now possible to generate started task JCL. Running as a STARTED TASK may be beneficial to some users as a STARTED TASK does not tie up a Batch Initiator; also, some sites have more flexibility to assign higher priority to Started Tasks compared to batch jobs.

The same Start Trace panel will be displayed as for a regular trace. The user should set the trace parameters as applicable.

However, instead of generating a version of the JCL that can be directly submitted, the procedure generates a job stream is copied to the specified user PROCLIB.

Unlike the regular version of the job stream, the generated job stream is re-executable. Each time it is submitted, it automatically increments the current trace number.

The generated job stream may be submitted using a job scheduler or using the normal /S command to start a started task.

Suppress Capture of Catalog Statistics

A new option has been added to the trace filters panel to suppress the collection of catalog statistics which normally runs at the end of a trace.

For some users, it may be desirable to suppress the automatic capture of the catalog statistics and to run this step explicitly at some later time.

Add User-Specified DD statement to Generated Job Stream

A user may have a requirement to include user specified DD statement(s) in the Tracer/cache job stream. For example

```
//HGBYPASS DD DUMMY
```

to suppress the functionality of the HOURGLASS product.

To activate this feature, add a member, containing the required DD card(s), named EXTRADD to *your.prefix*.CNTL

SQL Error Audit Trail Report

A new report has been introduced (25.10) that provides an audit trail of SQL errors. The user can now see each occurrence of an SQL error with the corresponding SQL detail and time stamp.

Alternatively, you can display the existing SQL error report (25.6) and DRILL on the SQL No or Program name to see the audit report for the selected SQL or Program.

Refer to the *EZ-DB2 R910 Reports Reference Guide* for further information.

SQL Access Path Report

A new report has been introduced (25.11) to display the access path information for all SQL in the workload.

Both version E and Version B access path will be displayed if available.

Refer to the *EZ-DB2 R910 Reports Reference Guide* for further information.

The following shows the EZ-DB2 R9.1 Main Menu:

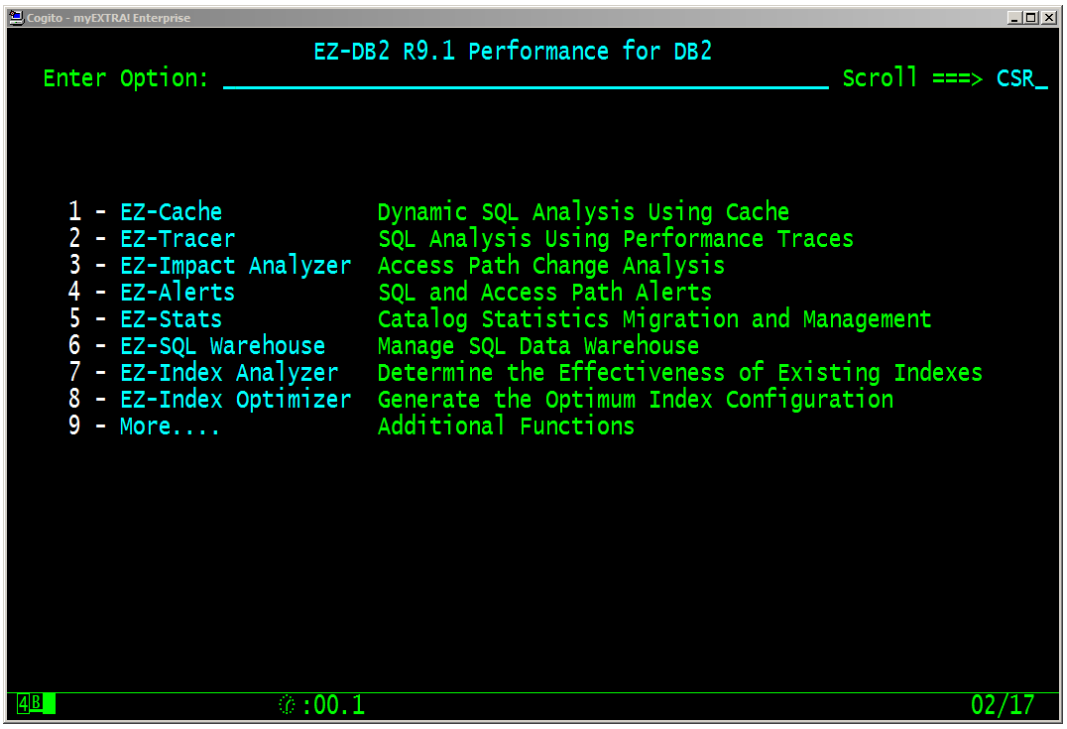


Figure 6 EZ-DB2 Main Menu

This User Guide describes the functionality of EZ-Cache and EZ-Tracer. For other EZ-DB2 components, refer to the applicable user guide.

Introduction

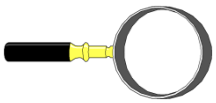
EZ-Cache

EZ-Cache**7** is a "sister" component of EZ-Tracer. Like EZ-Tracer, EZ-Cache provides the user with a comprehensive analysis of the SQL activity over a period of time. Before starting the Trace process, you must set the Trace Options as appropriate for the current Trace Workload

Whereas EZ-Tracer uses a low overhead DB2 Performance trace to capture the SQL activity, EZ-Cache collects the SQL activity by sampling the DB2 dynamic statement cache.

To capture activity from the Dynamic Statement Cache, EZ-Cache will start a MONITOR Trace for IFCID 316-318. If you already have such a trace active, then EZ-Cache can share it. Refer to Trace Controls later section 1.1 A MONITOR trace of IFCIDs 316-318 may run for many hours with minimal DASD required for the trace log files and minimal CPU overhead.

DB2 Version 10 introduces the ability to collect statement level statistics for both dynamic and static SQL statements. Monitor trace class 29 enables the collection of system-wide statement-level statistics for dynamic SQL statements in IFCID 0316, and for static SQL statements in IFCID 0401.



*To activate the DB2 Dynamic Statement Cache use the system parameters CACHEDYN and MAXKEEPD and the Bind parameter KEEPDYNAMIC(YES) . Refer to the **DB2 System Administration Guide** for further details about activating the Dynamic Statement Cache.*

You may use EZ-Cache to

- Trace ALL activity for a given subsystem within specified time intervals
- Trace activity for a given subsystem filtered by various selection criteria, such as:
    - Authids
    - Programs
    - DBNames and
    - Tables

Interim Summary Intervals are optional sub-divisions within the trace that allow the user to focus attention on particular parts of the trace. Data is retained for a specified number of the most recent intervals.

Continuous tracing enables tracing to run for an extended period. This works by seamlessly starting a fresh trace as each one ends. Combined with Summary intervals, this enables 24x7 monitoring. The

Duration of each trace and Summary Interval should be carefully chosen to provide the breakdown of data desired.

EZ-Cache provides the same summarization and consolidation of the SQL activity as EZ-Tracer. However, prior to DB2 10, it is limited to collecting data for Dynamic SQL only. If you wish to include Static SQL in your analysis then you should use EZ-Tracer, or a combination of both EZ-Tracer and EZ-Cache.

If you are at DB2 10 or higher, you can use EZ-Cache to capture both Dynamic and Static SQL.

Prior to DB2 10, to trace Dynamic SQL from the Cache and Static SQL, start two simultaneous Traces, i.e.:

- An EZ-Cache trace and
- An EZ-Tracer trace with Trace All, Static, Dynamic set to Static.

with unique job names for the initial trace step so they can run concurrently. All Dynamic SQL will then go to one Trace Workload and all the Static SQL to another.

Note that if tracing a SAP Z or PeopleSoft Z application, you must use the Dynamic Statement Cache trace. Refer to ERP Controls..

When starting an EZ-Cache trace, you can specify whether you wish to include SQL activity already in the Dynamic Statement cache prior to the EZ-Cache trace starting. Or you may choose to only include activity while the EZ-Cache trace is active. A further option allows you to take a "snap-shot" of the current contents of the Dynamic Statement Cache. Refer to Include Current Stats at Start option in section 1.1

Execution of the Trace job includes the following job steps:

- Collecting Trace data
- Summarizing and Consolidating the trace data
- Extracting System Catalog statistics from the system catalog for the applicable databases
- Generating the reports

Further information about the job steps can be found in Job Steps Summary and the Appendix of this User Guide.

EZ-Tracer

EZ-Tracer Z allows you to perform a trace to collect SQL trace data from the specified DB2 subsystem or all DB2 subsystems belonging to a DB2 Data Sharing Group. The trace runs as a stand-alone batch job utilizing the standard DB2 instrumentation facility. Before starting the Trace process, you must set the Trace Options as appropriate for the current Trace Workload

You may use this function to

- Trace ALL activity within specified time intervals for a given subsystem or all subsystems within a DB2 data sharing Group
- Trace activity filtered by various selection criteria for a given subsystem or all subsystems within a DB2 data sharing Group , such as:
    - Authids
    - Plans
    - Collections
    - Programs/Packages
    - Correlation
    - Locations
    - Connections

    - Workstation
    - UserID
    - Application Name
    - CICS Transactions
    - DBNames
    - Tables
- Limit Trace to only record instances of SQL that exceed threshold limits, such as
    - CPU
    - Elapsed Time
    - GetPages
- Trace activity for specific SQL statements only that have been pre-loaded into a Trace Workload using Load Application SQL From Catalog (See option 2.3) or Load Free Form SQL Text (See option 2.4)

Interim Summary Intervals are optional sub-divisions within the trace that allow the user to focus attention on particular parts of the trace. Data is retained for a specified number of the most recent intervals.

Continuous tracing enables tracing to run for an extended period. This works by seamlessly starting a fresh trace as each one ends. Combined with Summary intervals, this enables 24x7 monitoring. The

Duration of each trace and Summary Interval should be carefully chosen to provide the breakdown of data desired.

Trace Sampling allows the user to request a certain percentage of the system activity to be captured, rather than ALL activity as a means of reducing the overhead of collecting the data.

EZ-Tracer utilizes a DB2 Performance trace to collect the data. Note, however, that Performance Trace has a number of special Classes (30-32), which are reserved for local use. Using one of these Classes, a user can explicitly specify the set of IFCIDs required. ***EZ-Tracer*** *uses Performance Class 32 with a very selective list of IFCIDs, thereby minimizing the cost of running the trace.*

An EZ-Tracer Performance Trace by default will trace IFCIDs 58-61, 63-66,74,125,163,221,222,231,233).

This may include both Static and Dynamic SQL and provides more detailed information in the trace than a trace of Dynamic Statement Cache. Note however that if tracing a SAP® or PeopleSoft application, you must use the Dynamic Statement Cache trace.

Execution of the Trace job includes the following job steps:

- Collecting Trace data
- Summarizing and Consolidating the trace data
- Extracting System Catalog statistics from the system catalog for the applicable databases
- Generating the reports

Further information about the job steps can be found in Job Steps Summary and the Appendix of this User Guide.

Trace Workloads

EZ-Tracer and EZ-Cache traces are grouped together under Trace Workload names. You may include many traces (EZ-Cache and/or EZ-Tracer) in the same Trace Workload. Or you may create a new Trace Workload for each Trace.

To build up a picture of SQL activity over an extended period of time, it is more usual to add subsequent traces to the same workload. The new SQL activity will be consolidated with existing activity in the current workload.

The following panel shows the main menu for EZ-Cache. The menu for EZ-Tracer is similar.

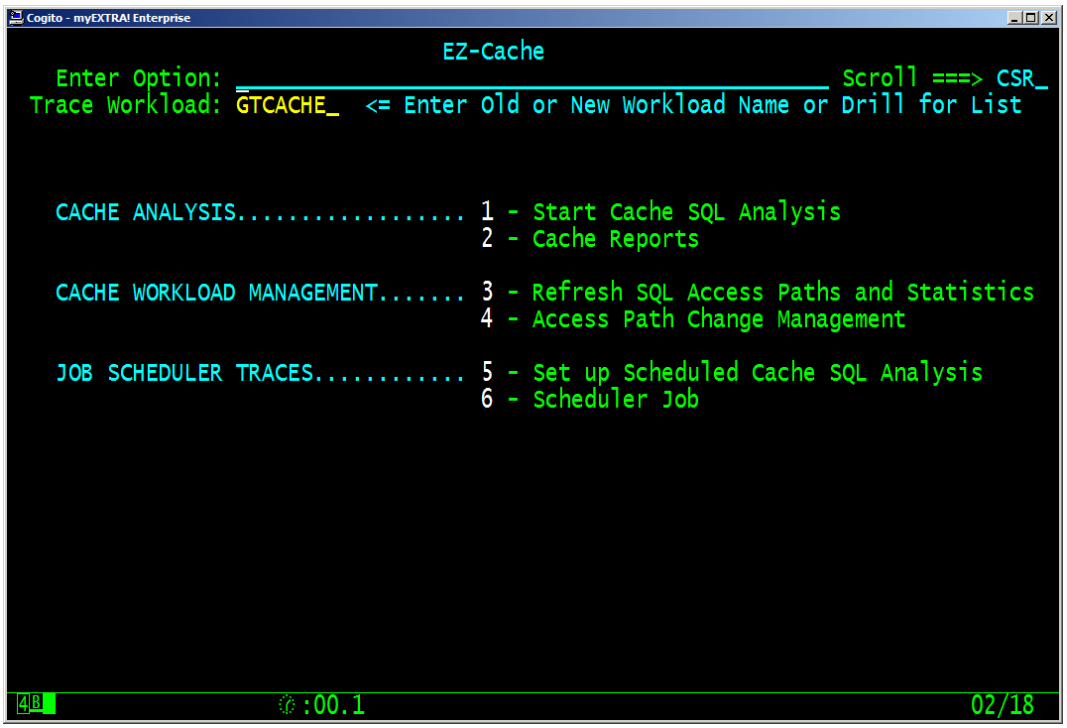


Figure 7 EZ-Cache Main Menu

To CREATE a new Workload, enter the new name in the Trace Workload Field.

A pop-up panel will be displayed confirming that you wish to create a new workload.

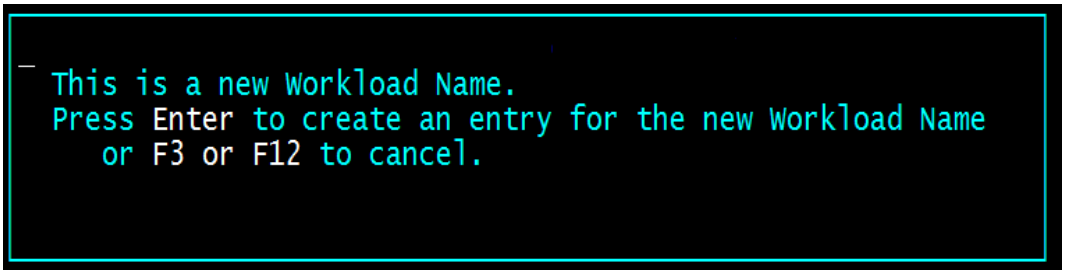


Figure 8 Create New Tracer Workload

Press <**Enter**> to confirm you wish to create a new workload or <PF3> or <PF12> to cancel.

When Creating a new Workload The settings for the new Workload are automatically copied from the Current Workload. The newly created Workload is set as the Current Workload

To display a list of existing workloads, you can DRILL <PF4> on the Trace Workload field.

You can filter the list of Workloads displayed by entering a MASK in the Trace Workload field before pressing the <ENTER> or DRILL <PF4> key.

To specify a MASK use % to match remaining characters. Use _ (underscore) to match any character in that position.

The following figure shows the Trace Workload Manager panel after the user has specified GT% in the Trace Workload field and pressed <ENTER>.

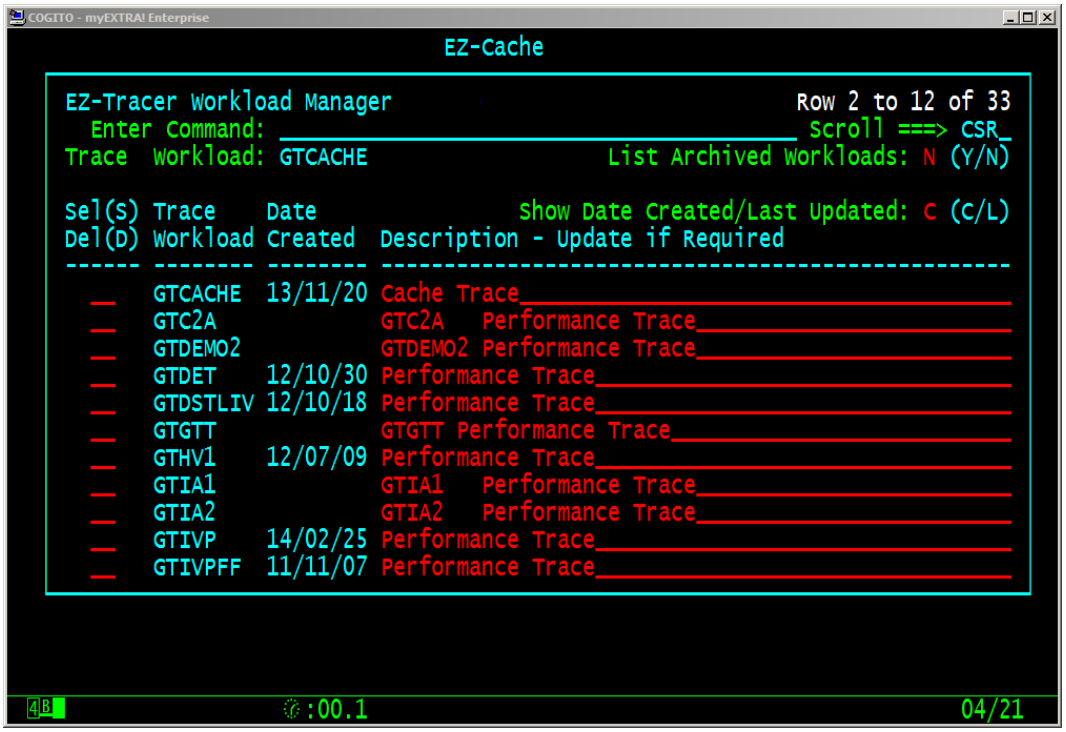


Figure 9 EZ-DB2 Workload Manager

After the initial install of the EZ-DB2 software, the default Trace Workload for the IVP (Installation Verification process) will already exist.

The panel fields are described below.

Trace Workload — The name of the current Trace Workload is displayed.

List Archived Workloads — Specify whether you wish to display the active or the archived workloads.

Y List the archived workloads. If you have specified a MASK, only the workloads matching the MASK will be displayed.

N List the active workloads. If you have specified a MASK, only the workloads matching the MASK will be displayed.

Show Date Created/Last Updated

You may toggle the display to show either the Date Created or the Date Last Updated for each workload.

C Show Date Created.

L Show Date Last Updated

You may SORT the workloads in the list by positioning the cursor in the Date Created/ate Last Updated column and pressing the <ENTER> key.

Sel(S)
Del(D)

You can specify a select value against an existing Trace Workload as follows:

S - Set this Workload as the current Trace Workload. **Note**: You can also set the current Trace Workload by over typing the Trace Workload name on any of the EZ-DB2 panels where the Trace Workload name appears.

D - Delete this Workload. A Job will be generated to delete the Workload datasets.

A - Archive the Workload. The Workload data is retained but the name is removed from the active workload list. To view the workload names that have been archived, change List Archived Workloads to Y.

R - Recall Workload. If a workload has not been referenced for a while, it's possible that the applicable datasets have been archived by DFSMSHSM. Specify R to send a Recall request to DFSMSHSM for all the workload datasets, If the workload has been archived it will also be restored from the list of archived workloads.

Note: You can use block commands, **AA**, **DD** and **RR** to archive, delete and restore multiple workloads.

Trace Workload

The Trace Workload name.
This field is sortable. Position cursor in this field and press the <enter> key.

Date Created / Date Last Updated

Either the Date Created or the Date Last Updated is displayed, depending on the setting of the Date Created/Last Updated toggle.
This field is sortable. Position cursor in this field and press the <enter> key.

Description

You may over type this field to enter some descriptive text that describes this Trace Workload.

EZ- Cache

1 EZ-Cache

Select option 1 to display the EZ-Cache Menu as shown in the following figure:

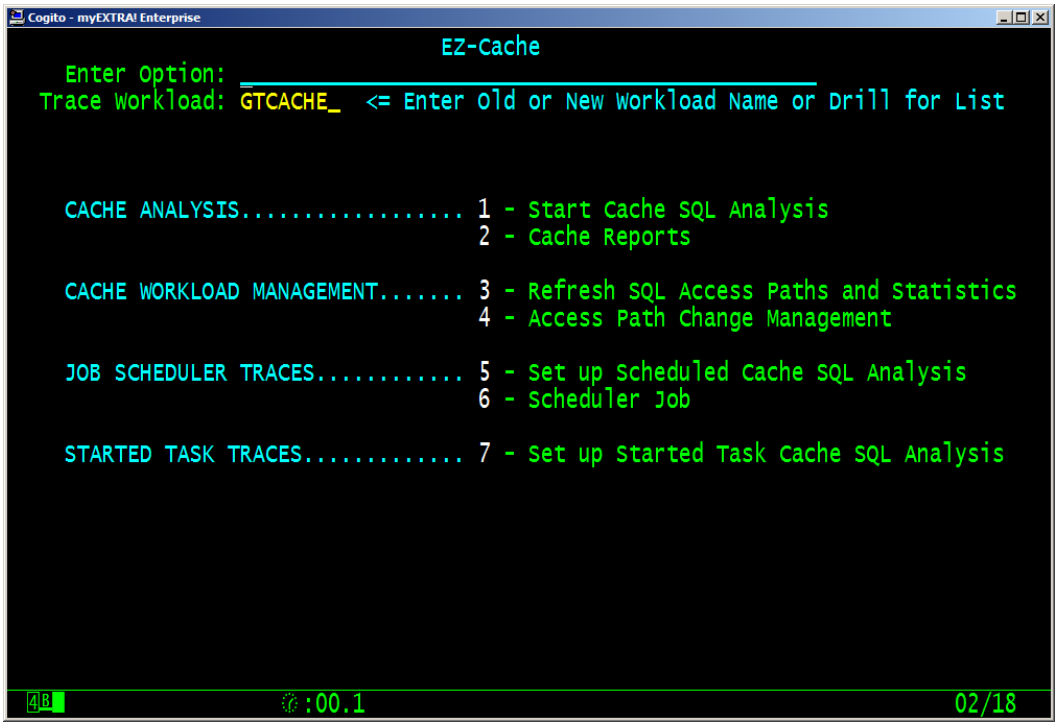


Figure 10 EZ-Cache Main Menu

The following options are displayed on this menu:

CACHE ANALYSIS

1.	**Start Cache SQL Analysis**	Set the Trace Options and Start the Trace.
2.	**Cache Reports**	Display the Trace reports menu.

CACHE WORKLOAD MANAGEMENT

3.	**Refresh SQL Access Paths and Statistics**	Generate the access paths for SQL if not generated at trace time, or refresh the access paths that were collected at trace time.
		Switch to the EZ-DB2 Impact Analyzer for the current trace workload.

4.	**Access Path Change Management**	

JOB SCHEDULER TRACES

5.	**Set up Scheduled Cache SQL Analysis**	This option is used to set up an EZ-Cache trace to be scheduled for automatic submission by a job scheduling system. Trace options are specified exactly as if submitting a manual trace.
6.	**Scheduler Job**	This option is used to display the Job to be submitted by the Job Scheduler. It is recommended this JCL be stored in a PDS used exclusively by Job Scheduler.

STARTED TASK TRACES

7.	**Set up Started Task Cache SQL analysis**	This option is used to set up a Cache Trace to be executed as a STARTED TASK.

These options are described in detail in the following sections.

1.1 Start Cache SQL Analysis

The Start Cache SQL Analysis option is used to start the process of collecting data from the DB2 Dynamic Statement cache.

The process will start a Monitor Trace for IFCIDs 316-318. However, if you already have such a trace active then EZ-Cache can share it. Refer to Trace Controls later in this section.

Execution of the Trace job includes the following job steps:

- Collecting Trace data
- Summarizing and Consolidating the trace data
- Extracting System Catalog statistics from the system catalog for the applicable databases
- Generating the reports

Further information about the job steps can be found in Job Steps Summary and the Appendix of this User Guide.

Before starting the Trace process, you must set the Trace Options as appropriate for the current Workload as described in the following section.

Specifying Trace Options

Select option 1.1 to display the Start Cache SQL Analysis Trace panel as shown in the following figure:

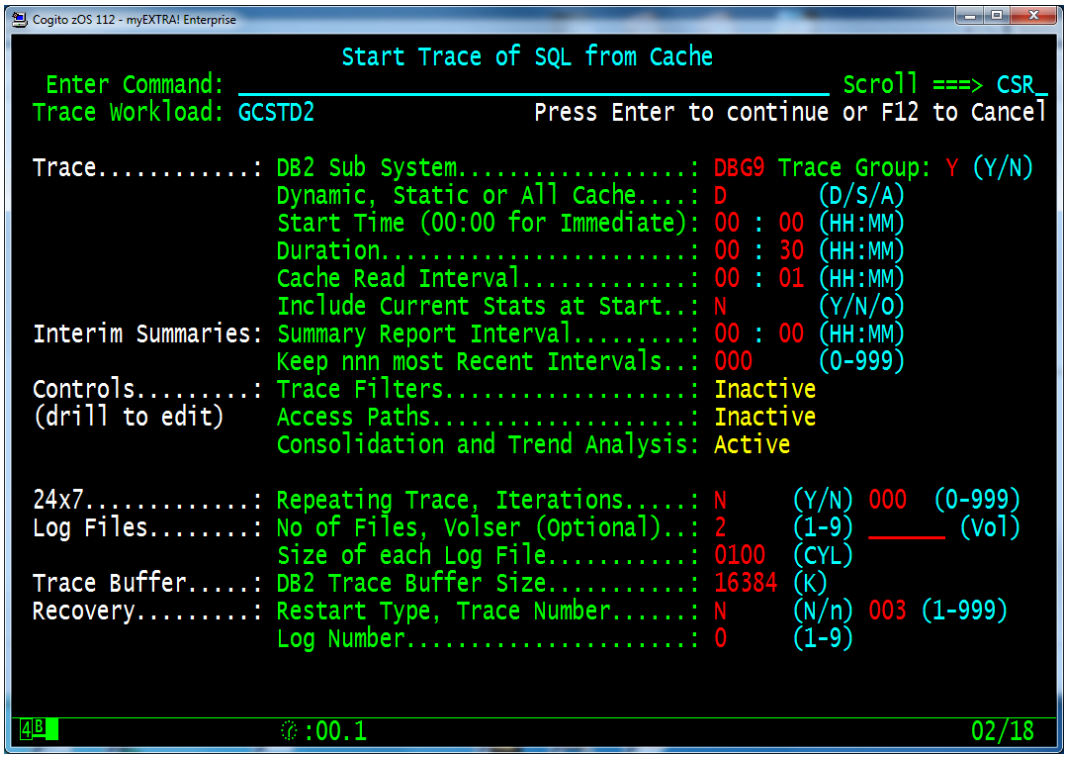


Figure 11 Start SQL Trace

The Trace parameter settings are set to the default values shown below when a user creates their first workload. The actual settings used for each workload are remembered. However, when a new workload is created the initial settings are the default values.

If you usually use settings which differ from the defaults as standard for most new workloads you may wish to set these as the default initial settings for your new workloads. You can do this by issuing the command **SET PROFILE** when in the command line. This will save the current settings shown on the panel as the default initial settings for your userid and these will be used as the initial settings whenever you create a new workload. You can also use the **COPY WORKLOAD** command to copy the parameters from an existing workload. Refer to the **EZ-DB2 Commands Reference** for further information.

The following fields may be specified on this panel.

Trace Workload — The Current Trace Workload name is displayed.

Trace

DB2 Sub System — Specify the name of the DB2 Subsystem to be traced or the name of a data sharing group.

The Sub system or Group name must have been previously activated in EZ-DB2, Refer to the EZ-DB2 Installation Reference Guide for further details.

Trace Group — If the DB2 subsystem name specified is a data sharing group name or a member of a data sharing group, this parameter will be displayed allowing you to specify whether you wish to trace just the group member of the entire group.

Y Trace activity in all the members of the data sharing group.
N Trace activity in this subsystem only.

Dynamic, Static or ALL Cache — This parameter may have one of the following values:

A Start Cache Trace capturing ALL SQL (both Static and Dynamic SQL).

S Start a Cache Trace capturing only Static SQL.

D Start a Cache Trace capturing only Dynamic SQL.

Note that prior to DB2 10, only dynamic SQL is captured in the Cache.

Start Time (00:00 for immediate) — Specify the Start Time of the trace. This is an optional parameter. If required, specify the start time of the trace in hh:mm format.

If this parameter is omitted, the trace starts immediately upon job submission.

You may set a start time on the following or subsequent day by specifying a Start Time greater than 24:00. For example, to start a trace tomorrow at 8:00 am, specify Start Time 32:00.

You do not have to specify a Start Time but you MUST specify Duration.

See also Trace single execution of Program parameter in Specify Trace Filters panel.

Duration

Specify the trace duration in hh:mm format. This determines the length of the trace.

See also the Trace Single execution of Program parameter in Specify Trace Filters panel.

Cache Read Interval

The interval on which to read data from Dynamic Statement Cache. Specify in hh:mm format. A maximum of 15 minutes is recommended to minimize data loss through SQL being flushed out of cache in a busy system.

Include Current Stats at Start

If cache is already active and statistics being accumulated before the trace is started, then the cache will hold performance data dating from when the statistics recording started. The following options may be specified:

Y — If you wish Statistics to be included for any Performance data in the Cache for SQL that was executed before the trace started.

N — If you want your trace to only collect data for SQL executed while the trace is active. In this case, any performance data relating to SQL executed before the trace started will be excluded. N is the default.

O — If you wish to only include statistics relating to SQL already in the cache at the start of the trace. In this case, there is no need to specify a Trace Duration. EZ-Cache will simply capture the current contents of the cache and then terminate.

Interim Summaries

Summary Report Interval

Specify the Summary Report Interval in hh:mm format. This is an optional parameter.

Optionally, specifies how often interim summary reports are written for the trace. This provides an extra level of granularity of the trace data and allows you to view trace data while the trace is still active. 00:00 means no interim summary reports. You can view the trace data for each interval as that interval completes. If you do not specify Interim Summary Report Interval, you are not able to view any trace data while the trace is active. You would have to wait until the end of the trace before you can view any data.

Keep nnn most Recent Summaries

Specify the number of most recent Interim Summary Reports to keep in *nnn* format.

For long traces, you may wish to only view the most recent Interim Summary Reports. Alternatively, you may wish to keep all the Interim Summary reports for the trace.

Example 1: Suppose you have specified 12:00 for the Duration, and furthermore you have specified Summary Report Interval 01:00 and Keep 012 most Recent Summaries. The trace will be broken down into 12 x 1 hour slots. You will be able to view the summarized data for each 1 hour Recent Summary interval while the trace is running. At the end of the 12-hour trace duration you will be able to see summarized data for the entire trace. You will also be able to DRILL on any of the 12 Summary Intervals to see summarized data for that period.

Example 2: Suppose you have specified 12:00 for the Duration, and furthermore, you have specified Summary Report Interval 00:15 and Keep 008 most Recent Intervals. In this case, you have a maximum of 2 hours of Summary Interval data available in 15 minute slots. Since the overall trace duration is greater than 2 hours, once the trace has run for more than 2 hours, the oldest Summary Interval data will be discarded. i.e., you will always have the most recent 2 hours of Summary Interval data available for online viewing. At the end of the 12-hour trace duration, you will be able to see the summarized data

for the entire trace, and you will be able to DRILL on specific Summary Interval data for the last 2 hours.

Controls

Trace Filters

The current Trace Filter status for the workload will be set to Active or Inactive.

Active - Trace Filters are active for this workload.

Inactive - Trace Filters are inactive for this workload.

DRILL <PF4> on the status to display the Trace Filters panel.

Refer to Trace Filter Controls later in this User Guide for further information.

Access Paths

The current Access Paths Controls status for the workload will be set to Active or Inactive.

Active - Access Paths Controls are active for this workload.

Inactive - Access Path Controls are inactive for this workload.

DRILL <PF4> on the status to display the Access Path Controls panel.

If you specify the access path controls here, then the access path information will be generated automatically as part of the Trace process and will be available for viewing in the Trace reports at the end of the trace.

Refer to Access Path Controls later in the User Guide for further information.

If you do not specify the controls here, you can still generate the access path information for the SQL in the Trace using Option 2.5 – Refresh SQL access paths and statistics.

Consolidation and Trend Analysis

The current Consolidation and Trend Analysis Controls status for the workload will be set to Active or Inactive.

Active - Consolidation and Trend Analysis Controls are active for this workload.

Inactive - Consolidation and Trend Analysis Controls are inactive for this workload.

DRILL <PF4> on the status to display the Consolidation and Trend Analysis Controls panel.

The panel allows you to specify Trend Analysis and Trace Consolidation criteria.

Refer to Consolidation and Trend Analysis Controls later in this User Guide for further information.

24x7

Repeating Trace, Iterations Y (Y/N) 000 (0-999)

Specify Y or N to indicate whether Repeating Trace is required. If you specify Y for Repeating Trace, then a new trace is automatically started at the end of each Trace Duration.

For example, if you specify Trace Duration 12:00 and Y for Repeating Trace, then a new trace will be started every 12 hours.

Specify the number of iterations in the format *nnn*, where *nnn* may be in the range 0-999. If you specify 0, the number of iterations is infinite. i.e., new traces will continue to be submitted at the end of each trace duration until you cancel the job using the STOP command as described below.

Note: The trace can be stopped by entering the command

/F jobname,STOP

in SDSF. It is not recommended that you CANCEL the job as trace data will be lost.

If you use the STOP command, then Tracer will end tidily, calling the background task to perform the summarization of the active log file at the time of termination. If you simply cancel the job, you will lose the data in currently active log file. You can however use the trace restart option to process the final log.

Log Files

No of files, Volser (optional)
1 (1-9) _____ (Vol)

No of Files: The number of Log Files to be used for the trace. The value may be in the range 1-9.

A minimum of three is recommended.

The default value is specified in the EZ-DB2 Configuration panel during the installation procedure.

VOLSER: Optionally, specify the VOLSER of the device on which the EZ-Tracer Log Files will be allocated.

If volser is specified, the generated DD statement will include the VOL=SER=volser parameter. If this parameter is omitted (e.g. SMS environments), the DD statement just includes the UNIT= parameter.

The value of unit is specified in the EZ-DB2 Configuration panel during the installation procedure.

Size of Each Log File

Cylinders: The space allocated for each Log File in cylinders.

Note: Log Files are used in rotation to pass raw trace data between the two run time components of EZ-Cache, XOPTRC1 and XOPTRC2. Switching to the next Log occurs either at the end of a Summary interval or when the current log file is full. Tracing a busy system will clearly produce more trace data and switch logs faster than a lightly loaded one. In these conditions, if logs are being switched more often than every 5 minutes, then we would suggest that the allocation be increased.

Regular trace:
Note: A busy system will collect about 5-10 cylinders of data per minute.

Dynamic Statement Cache trace:
The Cache is dumped at each Summary Interval and/or each Cache Read interval. Depending on SQL activity, up to 1 cylinder per megabyte of Cache is produced at each.

Log File Allocations:
Because of the way Tracer calculates the available space on the current log file, the files are allocated as a single extent and CONTIG. You should not alter this allocation. For SMS sites, it may also be necessary to specify MGMTCLAS=NORLSE to ensure that the files are pre-allocated to the correct size. See EZ-DB2 Configuration parameters in the installation procedure.

Trace Buffer

Trace Buffer Size

Specify the size of the buffer in K bytes into which the trace process receives trace data from DB2.

For DB2 V7 systems this should be 1024K (the maximum allowed)

For DB2 V8 systems and higher, a maximum of 16384K is allowed and the recommended minimum is 4096K. This is the default for V8.

When Tracing DB2 Groups large volumes of trace data are generated and the buffer size should be set no larger than 16384/n where n is the number of members in the Group.

If you see DSNW133I messages on the DB2 log while traces are executing, then this value should be increased.

Recovery

Restart Type, Trace Number

N (N/n) 001 (1-999)

Log Number

0 (1-9)

Different restarts are required depending on whether the failure occurred in:

1. Job 1 - The main trace job running XOPTRC1.

2. Job 2 - The trace consolidation jobs running XOPTRC2 that are submitted at the end of each interval or when a log file fills up and a log swap occurs.

3. Job 2 - In XOPTRC2 in the FINAL consolidation job submitted when the trace terminates.

4. Job 2 - In a step subsequent to XOPTRC2 in the FINAL consolidation submitted when the trace terminates.

5. JOB 3 - The final job that generates the EZ-Tracer Reports.

In (**1**) and (**3**) above, the trace may be restarted to shut the trace down cleanly, finish processing the final or current log file and submit Job 3 to process the reports data.

In (**2**) above no restart is possible and the data in the log file being processed will be lost unless you immediately cancel the trace, in which case it becomes like (1) above.

In (**4**) above the final job may be restarted to finish processing the trace data and submit Job 3 to process the reports data.

In (**5**) above Job 3 may be restarted to process the reports data.

To Restart:

Having corrected any problems, check the MVS log for Job 1 for the Trace Log File and Trace Number in use at the time of failure.

Search the job log for the LAST occurrence of messages:

DB200130 Start of Trace nnn
DB200120 End of Interval nnn Switching to LOG x

If there are no DB200120 messages on the log, then 1 is the Log Identifier.

Restart Type: Enter a numeric to identify the type of restart required. The following values may be entered:

1 To restart Job 1 for conditions (1) and (3)

2 To restart Job 2 for condition (4)

3 To restart Job 3 for condition (5)

The default value is N.

Log Id: Enter a numeric 1-9 to identify the current Log File.

Trace No: Enter nnnn, where nnnn is the current trace number.

The appropriate JCL will then be generated to restart at the correct point.

If the trace fails in XOPTRC2 then the log file processed at the time of failure may have been reused if the trace was still active. In that case the data is lost and recovery is not possible. Check the JCL for the log file in use by XOPTRC2 and the Trace Job Log messages to see if the log has been reused.

Once you have completed entry of the required parameters, press <**enter**> to validate. You are then prompted to press <**enter**> again to continue or **<PF12>** to cancel.

If you continue, the following pop-up panel is then displayed:

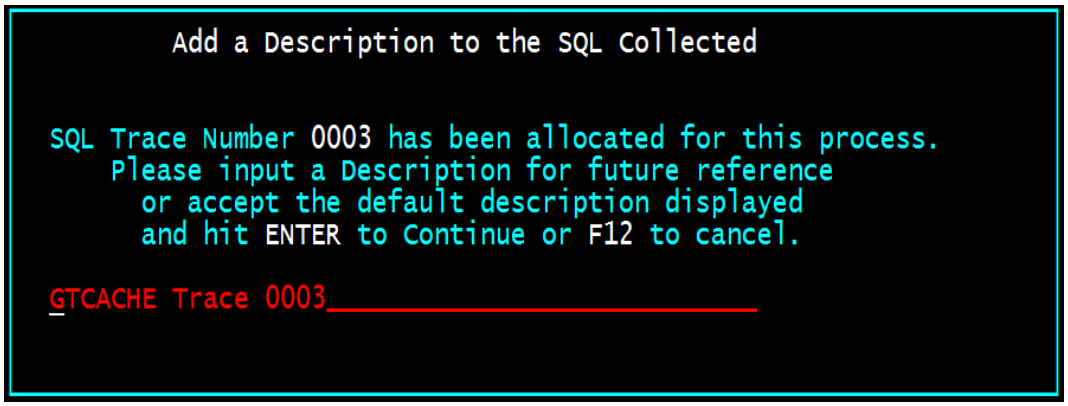


Figure 12 Add an SQL Trace Description

The panel displays the number of the next available SQL Trace File that will be created to contain the SQL collected for this Trace Workload.

You are prompted to enter a Description for this trace file. Enter a Description and press <**enter**> to display the generated job stream to start the DB2 trace.

- Type SUBMIT in the command line to submit the job for execution.
- Press the **<PF3>** key to return to the Start Cache Trace menu.

Trace Filter Parameters

The Trace Filter Parameters panel is displayed by drilling on the active/inactive indicator for Trace Filters Controls on the Start Cache Trace panel

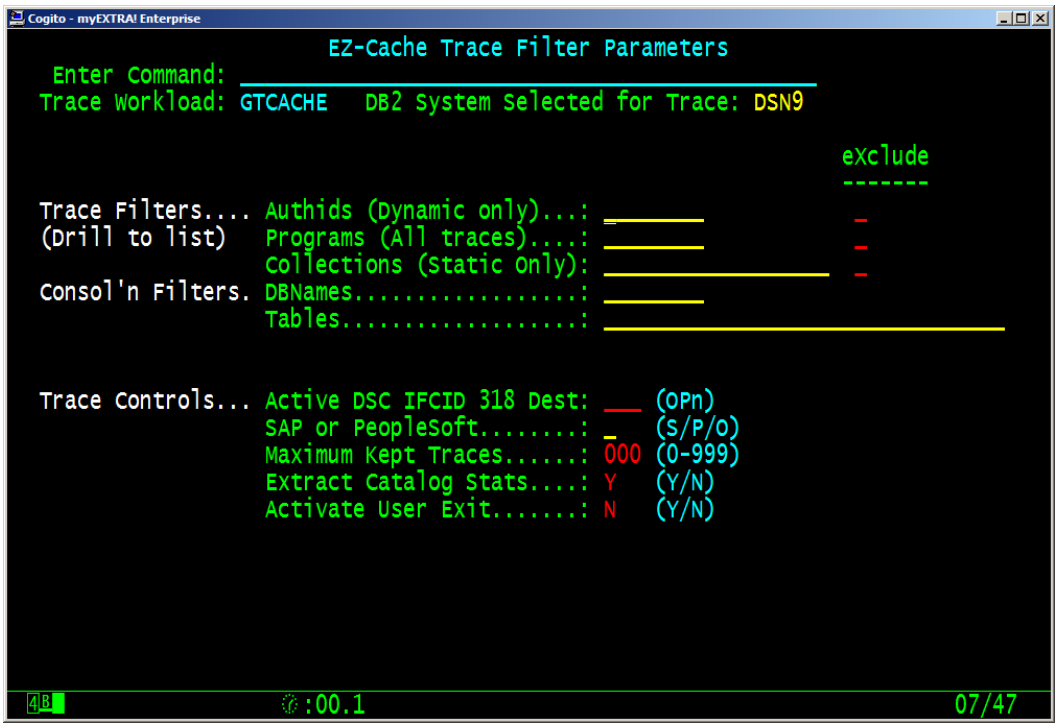


Figure 13 Cache Trace Filter Parameters

The following panel is displayed:

The following parameters may be entered on this panel:

Trace Workload — The name of the current Trace Workload.

DB2 System Selected for Trace — The DB2 subsystem ID of the DB2 system to be traced.

TRACE FILTERS

Authids (Dynamic Only) — This option applies to DYNAMIC SQL in the Statement cache only.

Limit the trace to specific AUTHID names. To enable a filter within DB2 you may include 1-8 full AUTHID names.

DB2 does not allow wild card entries but you may enter one or more AUTHID names with wild card '_' in any position and/or terminating with * to define a prefix and EZ-Cache will manually filter the trace data to include only those AUTHIDs that match.

E.g. Specify ABC* for anything starting ABC. Specify A_C* for anything starting with A in position 1 and C in position 3.

Use the DRILL Key <PF4> to display a list if you wish to enter more than one AUTHID filter.

Following the AUTHID field is an eXclude field which allows you to exclude these AUTHID's from the trace.

Specify **X** in this field to exclude these AUTHIDS from the trace.

Programs (All Traces)

Control the Trace by PROGRAM name. You may include 1-8 separate PROGRAM entries optionally with wildcard '_' and/or * to define a prefix.

E.g. Specify ABC* for anything starting ABC. Specify A_C* for anything starting with A in position 1 and C in position 3

Use the DRILL Key <PF4> to display a list if you wish to enter more than one PROGRAM filter.

Following the Programs field is an eXclude field which allows you to exclude these Programs from the trace.

Specify **X** in this field to exclude these Programs from the trace.

Collections (Static Only)

This option applies to STATIC SQL in the Statement Cache only.

Control the Trace by COLLECTION name. You may include 1-4 separate COLLECTION entries optionally with wildcard '_' and/or * to define a prefix.

E.g. Specify ABC* for anything starting ABC. Specify A_C* for anything starting with A in position 1 and C in position 3

Use the DRILL Key <PF4> to display a list if you wish to enter more than one COLLECTION filter.

Following the Programs field is an eXclude field which allows you to exclude these Collections from the trace.

Specify **X** in this field to exclude these Collections from the trace.

CONSOLIDATION FILTERS

DBNames

Filter the Trace to only collect information for the specified DBNAMES. You may specify up 1-8 DBNAMES in the Trace Filter parameters.

Use the DRILL Key <PF4> to display a list if you wish to enter more than one DBNAME filter. When you drill a list of available DBNAMES for the subsystem will be displayed from which you can select of deselect.

Tables

Filter the Trace to only collect information for the specified TABLES. You may specify up 1-8 TABLES in the Trace Filter parameters.

Use the DRILL Key <PF4> to display a list if you wish to enter more than one TABLE filter. When you drill a list of available Tables for the subsystem will be displayed from which you can select of deselect. If you have previously selected a DBNAME, then only the tables for the selected database will be displayed.

TRACE CONTROLS

Active DSC IFCID 318 Dest (OPn)

(OPn)

For traces of Dynamic Statement Cache, EZ-Cache will start a Monitor Trace for IFCID 318. If you already have such a trace active then EZ-Cache can share it, provided you specify the destination ID. This should be in the form OPn, where n is in the range 1 to 8. If this field is left blank, EZ-Cache will start its own trace.

You can determine the destination ID of an existing IFCID 318 trace using the DB2 command -Display Trace(*) .

SAP or PeopleSoft

(S/P/O)

Certain applications, typically SAP®, PeopleSoft and Other JDBC type applications make use of the SET CURRENT SCHEMA as a way of specifying the object qualifiers to be used by an application. This has the benefit of making the application code transportable. However, this presents a challenge for EZ-Tracer/Cache as the SET CURRENT SCHEMA may have been issued before the trace started, and hence Tracer/Cache has no way of knowing how to properly qualify the subsequent SQL.

The default action is for EZ-Tracer/Cache to use the current AUTHID as the qualifier; however, this may not be correct.

To resolve this issue, you may now specify the desired mapping of AUTHID to Object Qualifier.

On the Trace Filters panel the ERP controls option can now be used to specify the qualifier translation.

The following values may be specified:

S This is an SAP/R3® application.

P This is a PeopleSoft application

O This is some Other ERP application.

After entering the applicable value, DRILL on the field to display the AUTHID/Qualifier translation panel as shown in the following figure:

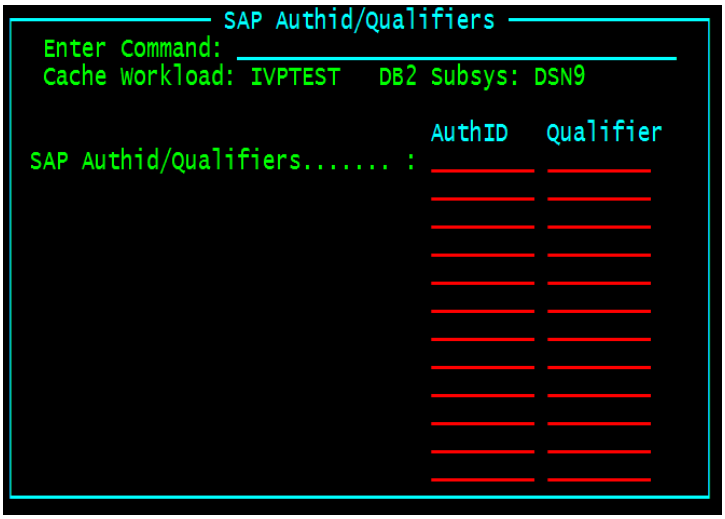


Figure 14 SAP Authid/Qualifiers

In this panel, you can enter up to 12 values for AUTHID and the associated default QUALFIER to be used.

Additional processing is required when copying the Catalog statistics for an SAP® workload to the analysis environment. Because of the dynamic nature of the SAP® environment, there is no guarantee that a table will be in the same Tablespace in the analysis environment as the live environment, and even corresponding Indexes may not have the same names in the Live and analysis environments. For further information, refer to the EZ-Stats User Guide. Appendix - Processing SAP® Workloads

It is also necessary to trace the dynamic statement cache rather than using the standard IFCID trace, as there is no guarantee that Tracer would collect the necessary PREPARE information required to capture the dynamic statement text.

It is not possible to mix SAP® and non-SAP® traces in the same Trace Workload. If you wish to analyze SAP® and non-SAP® workloads, you must create separate Trace workloads.

Maximum Kept Traces

If you wish to keep only a certain number of traces, then enter a value for the number of traces to be retained. A

value of 0 means all traces will be kept, otherwise the oldest traces will be deleted from the workload as new traces are generated with the nnn most recent retained.

Note that the workload totals and reports will still report values for all traces executed.

Deleting old traces frees up DASD and you lose the ability to drill into the deleted traces.

Extract Catalog Stats (Y/N)

Enter Y is you want the catalog statistics to be collected automatically for the databases reference by this workload at the end of the trace.

Note that this parameter works in conjunction with the **Refresh statistics** option on the access path controls. By default, the stats job that runs at the end of a tracer/cache run will only collect statistics for new databases that haven't previously been captured, unless you specify **Refresh statistics** = Y, in which case statistics for ALL databases will be collected.

If **Extract Catalog Stats** is set to N, then no statistics will be collected, whatever the value of **Refresh statistics**.

Activate User Exit (Y/N)

A user exit can be activated that allows user manipulation of the trace data before being written to the SQL log file for consolidation processing.

A sample user exit (EZTREXT1) is provided in the installation SRCLIB that modifies the AUTHID based upon a user supplied conversion table. For example:

AC% is converted to AC_GRP,
DC% is converted to DC_GRP
etc.

In this way, all possible AUTHIDS may be consolidated to a few departmental groups in a way that is meaningful to the user.

Specify Y to activate the user exit.

For further information about EZ-EXTREXT1 refer to the Appendix.

When you have completed entry of the required parameters press **<PF3>** key to save the parameters and to return to the Start Cache SQL Analysis Panel.

Access Path Controls

The Access Path Controls panel is displayed by DRILLING on the Active/Inactive indicator for Access Path Controls on the Start Cache SQL Analysis panel.

The following panel is displayed:

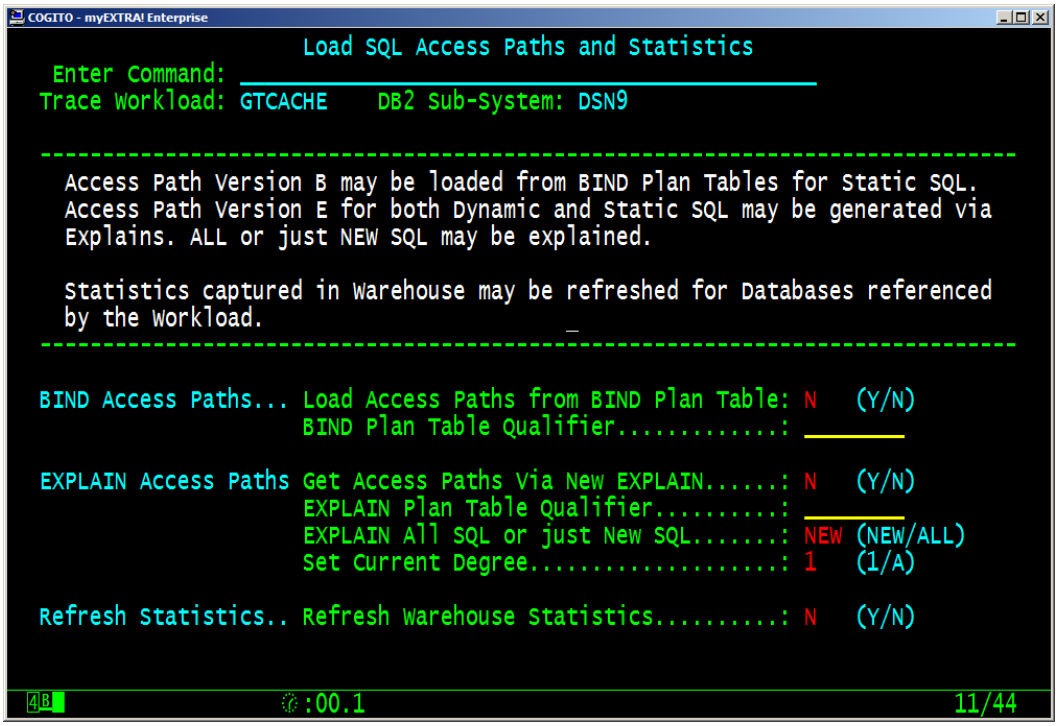


Figure 15 Specify Access Path Controls

Access Paths for SQL captured in the trace may be obtained by extracting from the applicable BIND PLAN Tables for STATIC SQL and /or explaining the SQL when the trace is terminated. By default, only NEW SQL where the Access Path is not already present will be explained, but you may opt for ALL SQL to be explained.

The BIND PLAN Table access path will generate Version B of the access Path data; The EXPLAIN process will generate Version E of the Access Path Data.

Note that if you do not choose to generate access paths when the trace terminates, you can subsequently add the access paths for the SQL in the trace using Options - 1.3 or 2.5 – Refresh access paths and Statistics.

The following parameters may be specified-

BIND Access Paths

Load Access Paths From BIND Plan Table

Specify whether you wish to get access paths from Plan table(s) used at BIND Time.

Y - Specify Y If you want to get the access paths from the Plan Table used at BIND time. You must have specified EXPLAIN(YES) when performing the BIND. This will load or refresh the current Version B access path for each SQL in the workload.
SQL not found in the BIND Plan Table will be Explained now (if you have also specified Get Access Paths via New Explain). This will load or refresh the current Version E access path for each SQL in the workload

N - Specify N if you do not want to get access paths from a BIND Plan table.

BIND Plan Table Qualifier

Specify the applicable qualifier(s) to be used to identify the BIND Plan table(s) to be used.

DRILL <PF4> be presented with a list of available PLAN_TABLEs from which you can select up to 15.

EXPLAIN Access Paths

Get Access Paths via New Explain

Specify whether you wish to get access paths by explaining the SQL.

Y - Specify Y to Explain the SQL using the specified PLAN table. This will load or refresh the current Version E access path for each SQL in the workload.

N - Specify N if you don't want to explain the SQL.
This is the default value.

EXPLAIN Plan Table Qualifier

Specify the applicable qualifier to be used to identify the PLAN_TABLE to be used for the explain.

DRILL <PF4> to be presented with a list of available PLAN_TABLEs from which you can select the applicable qualifier

EXPLAIN All SQL or just New SQL

NEW - Only new SQL where the access path is not already present will be explained. This is the default value.

ALL - Explain all SQL.

Set Current Degree

A — Set CURRENT DEGREE to ANY, prior to issuing the explains. This will allow parallelism, if parallelism is enabled for the subsystem.

1 — Leave CURRENT DEGREE as default 1, to disable parallelism.

Refresh Statistics

Refresh Warehouse Statistics

(Y/N)

At the end of each trace, statistics are captured to the statistics repository (warehouse) for each new database encountered. These are the statistics that are displayed when a user drills to a database object (e.g. table or index) from the access path display.

The access paths may be generated at the end of the trace, or explicitly using option EZ-Cache option 1.3 or EZ-Tracer option 2.5 to refresh the access paths.

By default, the statistics are captured for a given workload only the first time that the database is encountered. This means that it is possible for the statistics to become *stale*, and possibly not correspond to the statistics that were present at the time the access path was generated.

The Refresh Catalog statistics option in EZ-Cache and EZ-Tracer may be used to refresh the statistics each time the access paths are generated.

Y — Specify Y to force all statistics to be refreshed for the workload at the end of the trace.

N — The statistic will not be refreshed. They will only be extracted the first time the database is encountered for this workload/

Note the option to EXPLAIN all SQL or just NEW SQL. To ensure that all access paths for all SQL are in synch with the statistics, then you should EXPLAIN ALL SQL and refresh the statistics.

When you have completed entry of the required parameters press **<PF3>** key to save the parameters and to return to the Start Cache SQL Analysis Panel.

Consolidation and Trend Analysis Controls

The Consolidation and Trend Analysis Controls panel is displayed by DRILLING on the Active/Inactive indicator for Consolidation and Trend Analysis Controls option on the Start Cache SQL Analysis panel.

The panel is used to specify the Consolidation and/or the Trend Analysis data collection criteria if required.

The following panel is displayed:

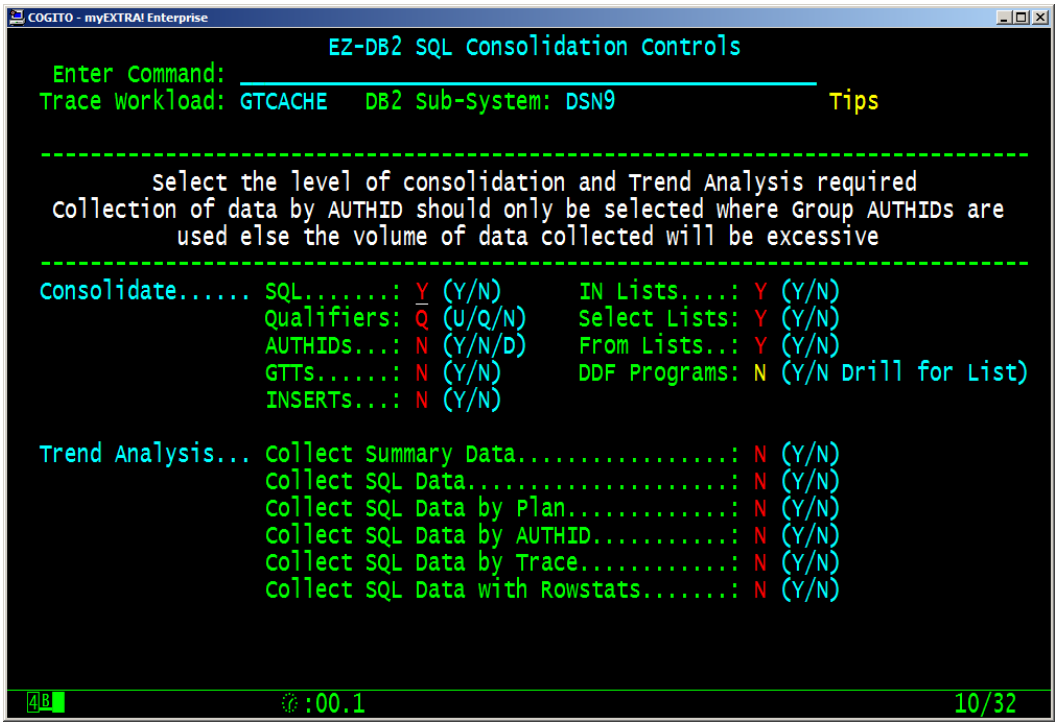


Figure 16 Specify Consolidation and Trend Analysis

For consolidation, the panel is used to specify whether SQL consolidation is active, and if so, to further specify the Consolidation actions to be taken.

A Trend Analysis database can now optionally be created for each DB2 subsystem being traced. If created, EZ-Cache will populate the database with relevant performance data at the end of each trace process. Refer to the EZ-DB2 Installation Reference Guide for further information regarding creation of the Trend Analysis Database.

The database is a standard DB2 database that can be accessed by user written programs. It is particularly useful for users wishing to perform trend analysis on other platforms. For example, Microsoft Windows7 users can use DB2 Connect to access the database directly from Excel7. The data available allows users to plot the performance characteristics of SQL by Plan, Program, Authid etc. over time, thereby making it easy to identify aberrations in the performance of particular statements, or to correlate variations in performance e.g., total/average CPU, Getpages etc. with particular events, such as application changes or systems maintenance such as Reorg or Runstats. The contents of the database are documented in the EZ-DB2 Tables Reference Guide. This panel is used to define the data that is to be written to the Trend Analysis database.

The Consolidation and Trend Analysis Controls settings are set to the default values shown below when a user creates their first workload. The actual settings used for each workload are remembered. However, when a new workload is created the initial settings are the default values.

If you usually use settings which differ from the defaults as standard for most new workloads you may wish to set these as the default initial settings for your new workloads. You can do this by issuing the command SET PROFILE when in the Consolidation and Trend Analysis Controls panel. This will save the current settings shown on the panel as the default initial settings for your userid and these will be used as the initial settings whenever you create a new workload. Refer to the EZ-DB2 Commands Reference for further information about the SET PROFILE command.

The following parameters may be entered on this panel:

Consolidate

SQL

The following values may be entered for this parameter.

Y Consolidate Equivalent Dynamic SQL.

N Do not Consolidate Dynamic Equivalent SQL.

Dynamic SQL may appear many times in a Workload for the same Program and Statement number where the only difference in syntax is the value of an explicitly defined literal.

E.g.: WHERE CUSTNUM = 12345 is equivalent to
WHERE CUSTNUM = 56789

Capturing all the distinct versions of the same SQL will result

in much larger Workloads and make it difficult to measure the true total and average costs of the SQL Statement as the costs will be spread over all the distinct versions. Consolidating these SQL into a single version will enable you to see the cost of all executions of the SQL in one. The SQL text captured will be the first version encountered.

The default value is Y.

Qualifiers

U Enter U to consolidate UNQUALIFIED all equivalent SQL using different qualifiers for the same table names. This is the default and most effective form of consolidation which will consolidate equivalent SQL where the ALIAS, VIEW or TABLE qualifiers are not equal. For example: two SQL referencing ALIAS1.TableName and ALIAS2.TableName, where Tablename is the same, will be treated the same

and reported as one SQL. Likewise, the SQL for a package bound twice with different qualifiers.

Q Enter Q to consolidate QUALIFIED all equivalent SQL using different qualifiers for the same table names. For example: where a program is bound into two plans with different qualifiers then all the SQL will be reported as two different SQL, each with its own qualifier.

N Enter N to NOT perform any qualifier processing prior to consolidation. Two otherwise identical SQL, one coded fully qualified and one unqualified, will be reported as two different SQL.

The default value is N.

AUTHIDs

Y Enter Y to consolidate all SQL for AUTHIDs. There will be no breakdown of SQL Statistics by AUTHID.

D Enter D to accumulate trace data by AUTHID for Distributed Dynamic SQL only. Typically, GROUP AUTHIDs will be in use. SQL Statistics will be broken down by AUTHID for all DDF activity only.

N Enter N to accumulate trace data by AUTHID and allow full AUTHID reporting. Note This option should only be taken where GROUP AUTHIDs are extensively in use else the volume of data may be prohibitive.

The default value is D.

GTTs

Enter Y to enable consolidation of Global Temporary Table Names. In some applications, particularly Micro Strategy®, the same table may be defined many times using different table names. This results in many variations of the same SQL as each execution appears to reference different tables.

In other applications, the same table name may be used for what are in fact very different definitions. In either case enabling consolidation will result in the table definitions being analyzed and a unique internally generated name allocated to each unique table definition. The SQL that reference the tables are then updated to use the newly generated name. All equivalent SQL that reference each unique table will then be consolidated into one.

The default value is N.

INSERTS	Enter Y to enable consolidation of single row INSERT statements. INSERTs for the same table are consolidated when there are no SELECT sub-clauses. The default value is N.
IN LISTs	Enter Y to consolidate equivalent SQL with differing numbers of entries in IN LISTs. The default value is Y.
Select Lists	Enter Y to enable consolidation of columns in Select Lists. SELECT COL_A, COL_B and SELECT COL_B, COL_A are then treated as being the same SQL and consolidated into one. The default value is Y.
From Lists	Enter Y to enable consolidation of tables in From Lists. FROM TABLE_A, TABLE_B and FROM TABLE_B, TABLE_A are then treated as being the same SQL and consolidated into one. The default value is Y.
DDF Programs	Enter Y to consolidate SQL for specific DDF Program Names into the Program Name DDF. Typically, DDF program names are not meaningful as they are simply SQL Drivers so it does not help to see SQL execution frequencies and statistics for each version of the drivers. Consolidation reduces the volume of data collected and provides more meaningful reports. The default value is N. **Note**: You may DRILL on this field to see and update a list of DDF program names that will be consolidated as shown in the following figure:

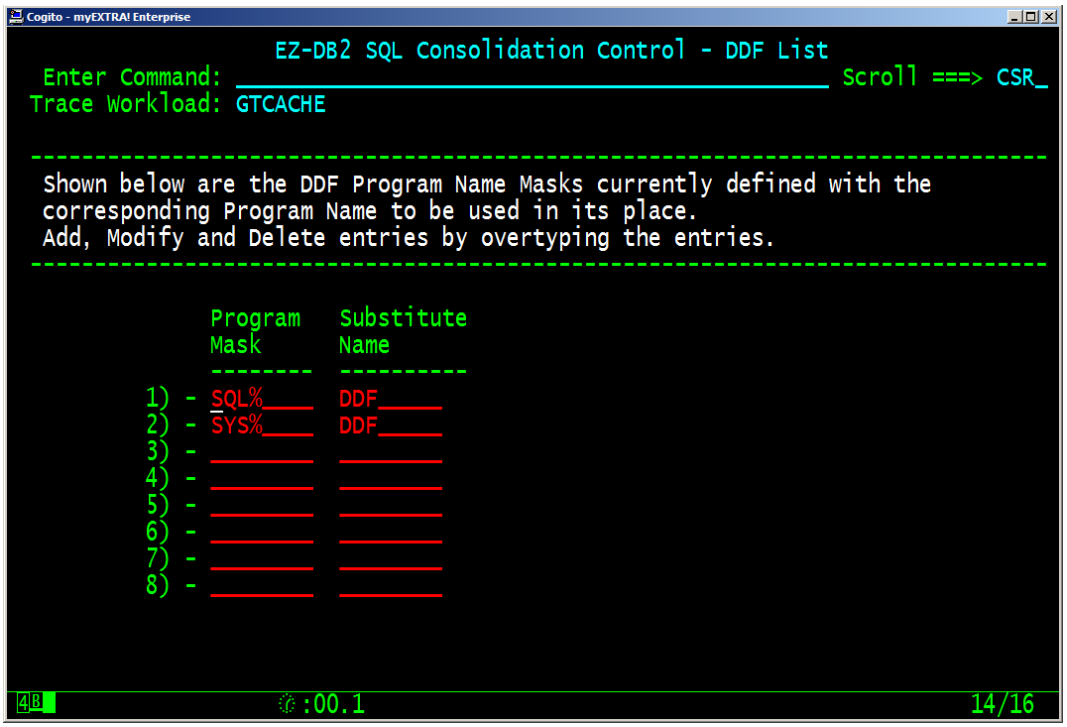


Figure 17 DDF Program consolidation list

Program name prefixes may be specified e.g. SYSN%.

Trend Analysis

Collect Summary Data

Enter Y to update the Trend Analysis database with Summary data for this trace.

The default value is N.

Collect SQL Data

Enter Y to update the Trend Analysis database with SQL data for this trace.

Enter D to delete the existing SQL Trace data for this Workload.

The default value is N.

Collect SQL Data by Plan

Enter Y to update the Trend Analysis database with SQL data by Plan for this trace.

Enter D to delete the existing SQL Trace data by Plan for this Workload.

The default value is N.

Collect SQL Data by Authid	Enter Y to update the Trend Analysis database with SQL data by Authid for this trace. Enter D to delete the existing SQL Trace data by Authid for this Workload. The default value is N.
Collect SQL Data by Trace	Enter Y to update the Trend Analysis database with SQL data by Trace for this trace. Enter D to delete the existing SQL Trace data by Trace for this Workload. The default value is N.
Collect SQL Data with Rowstats	Enter Y to update the Trend Analysis database with SQL data with Rowstats for this trace. Enter D to delete the existing SQL Trace data with Rowstats for this Workload. The default value is N.

Job Step Summary

The generated job contains the following internal job steps:

Job Step	Description
ALLOCDS	Allocate required datasets
ALOCRPTV[1]	Allocate VSAM Report datasets
RUNVSAM3[1]	Initialize VSAM Report datasets
ALOQSQLV	Allocate EZ-DB SQL VSAM Work Datasets
XOPTRC1	This is the main trace job that collects the SQL trace data from the applicable DB2 subsystem.
XOPTRC2	Analyze SQL Trace. This Step is submitted automatically at the end of the Trace Interval (or at the end of each Summary interval where Summary interval has been specified) or when a Log switch occurs due to the log full condition.
	The following job steps are executed after the final iteration of XOPTRC2
RUNPATH	Generate access path data if access path controls have been specified
SUMMARY	Accumulate Workload Summary. The data for the entire trace is made available to the Online Reporting panels.
COMPRESS	Compress CATSTATS dataset in preparation for running RUNSTATD

[1]First time for this workload only

RUNSTATD	Extract System Catalog statistics for Traced databases to the statistics Warehouse. Refer to Refresh Catalog statistics option.
DELWORK	Delete EZDB2 trace VSAM datasets
DELLOGF	Delete EZDB2 trace Log datasets
REPSUB	Submit report jobs
DELGRPTS	Existing report files for the workload are deleted before they are recreated by the next step.
REPORTT	Generate Reports for Tracer - Current Trace
REPORTW	Generate Reports for Tracer - Workload Reports

For further details of these job steps refer to the Appendix.

Manually Stopping the Trace

It is possible to terminate the trace process at any time by entering the following modify command from within SDSF.

/F *jobname*,STOP or

/F *jobname*,SHUTDOWN

Where *jobname* is the job name as applicable for the trace job.

It is recommended that this command be used if you wish to terminate a trace before the expiration of the trace duration. When terminated using this command, the trace will end cleanly and the summarization and reporting functions will be executed for the currently active log file.

If you cancel the trace with a CANCEL or PURGE command, the data in the last logfile will be lost.

See also Trace Restart option in Start Cache SQL Analysis

Manually Extending the Trace duration

Occasionally, a user may wish to extend the duration of a trace that is already in progress.

To extend the trace time enter the command

/F *jobname*,ADD,*nnnn*

Where *jobname* is the jobname of the trace job and *nnnn* is the number of minutes to be added to the trace duration. Leading zeroes are mandatory, for example

/F CLGXB831,ADD,0010

to add 10 minutes to the trace time. *nnnn* should be in the range 1 – 1440 (24 hours).

The command may be entered any time after the trace has started and may be repeated if further extensions are required any time before the trace ends.

1.2 Cache Reports

Select option 1.2 to display the Cache Analysis Reports menu. The following panel is displayed:

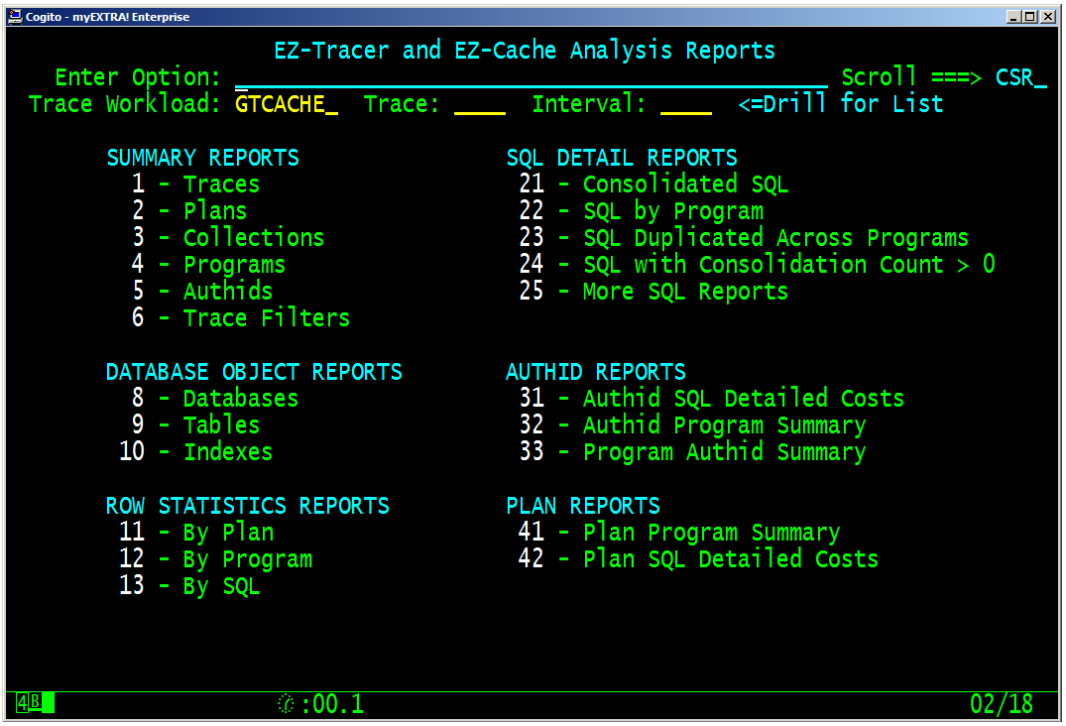


Figure 18 Trace reports menu

A comprehensive set of reports is available to provide detailed analysis of the Cache Trace workload. These reports are equivalent to the reports produced when using EZ-Tracer to generate Trace Analysis.

Refer to the EZ-DB2 Reports Guide for further information.

1.3 Refresh SQL Access Paths and Statistics

This option is used when you wish to add or refresh access paths and catalog statistics for SQL in an existing trace workload.

Access Paths for SQL captured in the trace may be obtained by explaining the SQL when the trace is terminated. By default, only NEW SQL where the Access Path is not already present will be processed but you may opt for ALL SQL to be explained. This will generate Version E of the Access Path Data. Refer to the Start Cache SQL Analysis option Access Path Controls.

If you do not choose to generate access paths when the trace terminates, or you wish to refresh the access paths and/or catalog statistics, you can do so using this option.

In addition, you may use this option to load or refresh the access paths for static SQL from the applicable BIND PLAN table. This will populate Version B of the access path.

The following panel is displayed:

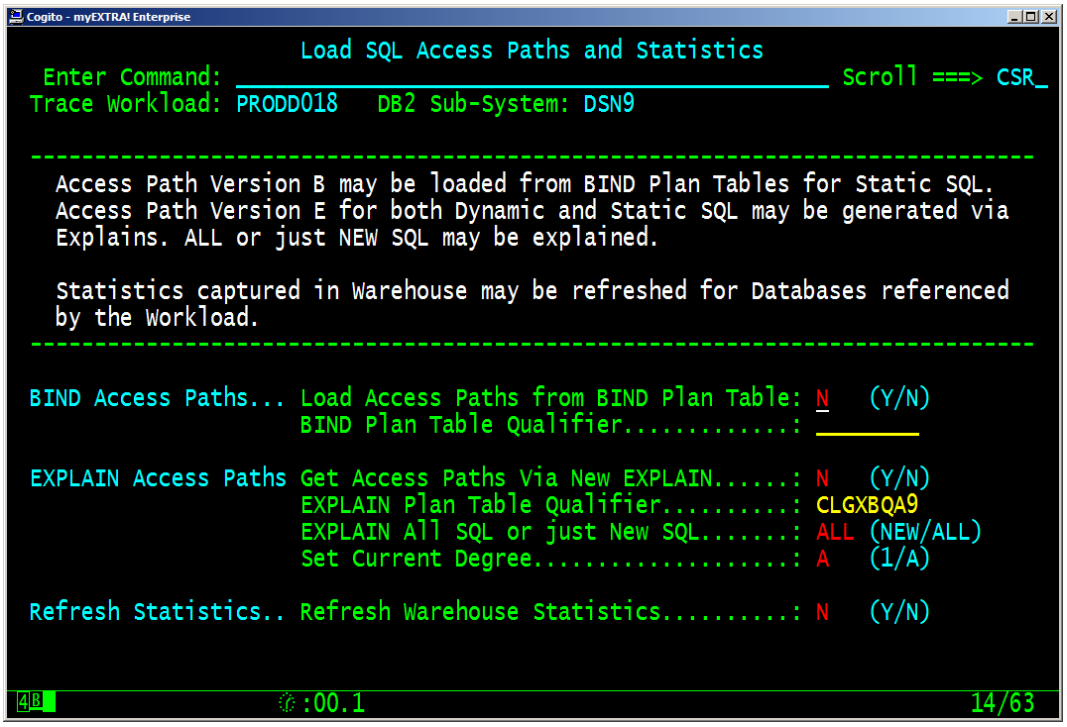


Figure 19 Refresh SQL access paths and statistics

The following parameters may be specified-

BIND Access Paths

Load Access Paths From BIND Plan Table

Specify whether you wish to get access paths from Plan table(s) used at BIND Time.

Y - Specify Y If you want to get the access paths from the Plan Table used at BIND time. You must have specified EXPLAIN(YES) when performing the BIND. This will load or refresh the current Version B access path for each SQL in the workload.

SQL not found in the BIND Plan Table will be Explained now (if you have also specified Get Access Paths via New Explain). This will load or refresh the current Version E access path for each SQL in the workload

N - Specify N if you do not want to get access paths from a BIND Plan table.

BIND Plan Table Qualifier

Specify the applicable qualifier(s) to be used to identify the BIND Plan table(s) to be used.

DRILL <PF4> be presented with a list of available PLAN_TABLEs from which you can select up to 15.

EXPLAIN Access Paths

Get Access Paths via New Explain

Specify whether you wish to get access paths by explaining the SQL.

Y - Specify Y to Explain the SQL using the specified PLAN table. This will load or refresh the current Version E access path for each SQL in the workload.

N - Specify N if you don't want to explain the SQL.

This is the default value.

EXPLAIN Plan Table Qualifier

Specify the applicable qualifier to be used to identify the PLAN_TABLE to be used for the explain.

DRILL <PF4> to be presented with a list of available PLAN_TABLEs from which you can select the applicable qualifier

EXPLAIN All SQL or just New SQL

NEW - Only new SQL where the access path is not already present will be explained. This is the default value.

ALL - Explain all SQL.

Set Current Degree

A — Set CURRENT DEGREE to ANY, prior to issuing the explains. This will allow parallelism, if parallelism is enabled for the subsystem.

1 — Leave CURRENT DEGREE as default 1, to disable parallelism.

Refresh Statistics

Refresh Warehouse Statistics

(Y/N)

At the end of each trace, statistics are captured to the statistics repository (warehouse) for each new database encountered. These are the statistics that are displayed when a user drills to a database object (e.g. table or index) from the access path display.

The access paths may be generated at the end of the trace, or explicitly using option EZ-Cache option 1.3 or EZ-Tracer option 2.5 to refresh the access paths.

By default, the statistics are captured for a given workload only the first time that the database is encountered. This means that it is possible for the statistics to become *stale*, and possibly not correspond to the statistics that were present at the time the access path was generated.

The Refresh Catalog statistics option in EZ-Cache and EZ-Tracer may be used to refresh the statistics each time the access paths are generated.

Y — Specify Y to force all statistics to be refreshed for the workload at the end of the trace.

N — The statistic will not be refreshed. They will only be extracted the first time the database is encountered for this workload/

Note the option to EXPLAIN all SQL or just NEW SQL. To ensure that all access paths for all SQL are in synch with the statistics, then you should EXPLAIN ALL SQL and refresh the statistics.

Press **<enter>** to display the generated job stream to generate the Access Path data.

Press **<enter>** to display the generated job stream to generate the Access Path data.

- Type SUBMIT in the command line to submit the job for execution.

- Press the **<PF3>** key to return to the EZ-Cache menu.

If you have received Explain failures the job will terminate with RC=4 and you should examine the log file for further information about the statements that have not been explained successfully.

Job Step Summary

The generated job contains the following internal job steps:

Job Step	Description
ALLOCDS	Allocate required datasets
RUNPATH	Generate Access Path data for all SQL in Workload[2]
COMPRESS	Compress the STATS file[3]
RUNSTATD	Extract the catalog statistics to the STATS file[3]

For further details of these job steps refer to the Appendix.

[2] IF Get Access Paths is YES
[3] IF Refresh Statistics is YES

1.4 Access Path Change Management

It is possible to perform Impact Analysis directly on Trace Workloads using the EZ-Impact Analyzer product. Select Option 1.4 to display the Impact Analyzer for Trace Workloads menu as shown in the following figure:

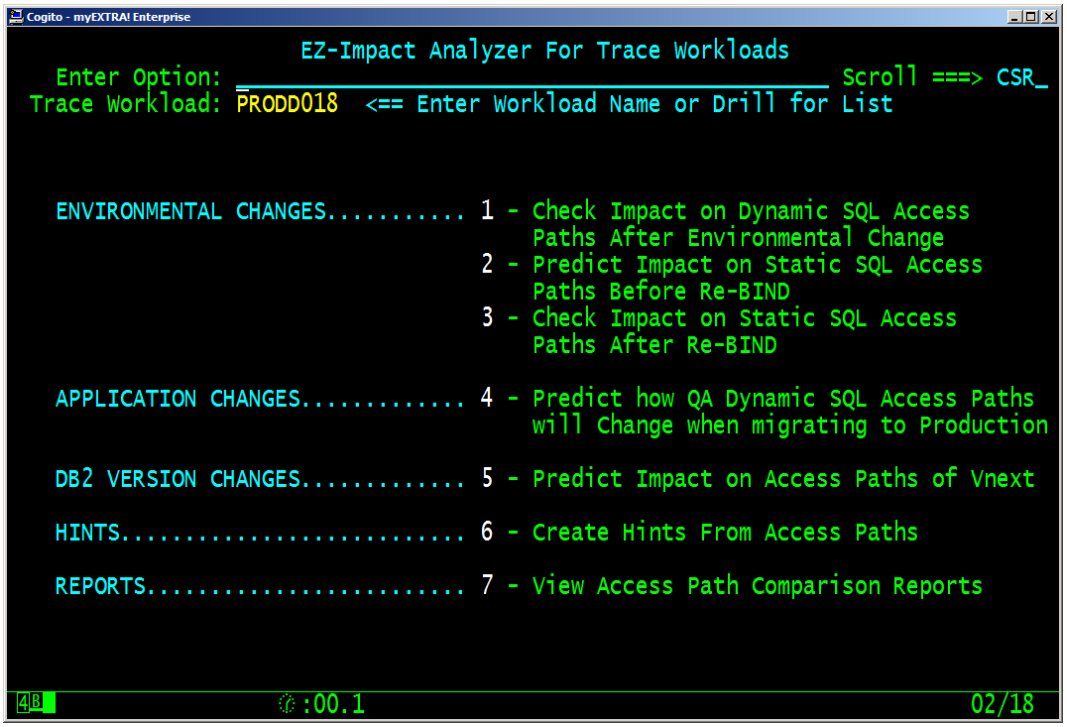


Figure 20 Impact Analyzer for Trace Workloads

The functions on the Analyze Traced SQL menu are divided into the following groupings:

Environmental Changes

1- For Dynamic SQL, Check the impact of an environmental change by comparing a new explain with the existing (Version E) access path.

2- For Static SQL, predict the impact of an environmental change by comparing a new explain with the existing (Version B) access path.

3- For Static SQL, check the impact of an environmental change after a re-bind by comparing the new bind access path with the existing (Version B) access path.

Application Changes

4- For Dynamic SQL, predict the impact of migrating to Production. Compare new Explain in the target subsystem with the existing (Version E) access path.

DB2 Version Changes	**5**-	For Static and/or Dynamic SQL, predict the impact of migrating to DB2 Vnext. Perform new explain in the Vnext environment and compare with existing Version E or Version B access paths for Dynamic and/or Static SQL respectively.
Hints	**6 -**	Create Hints from Access Paths
Reports	**7 -**	Display the Reports menu

Refer to the EZ-Impact Analyzer User Guide for further information. Note you must be licensed for the EZ-Impact Analyzer product.

1.5 Set Up Scheduled Cache SQL Analysis

It is possible to set up an EZ-Cache trace to be scheduled for automatic submission by a job scheduling system. For example, you may wish to submit a 5-minute cache trace every hour on the hour.

Select option 1.5 to display the Start Cache trace panel as shown in the following figure:

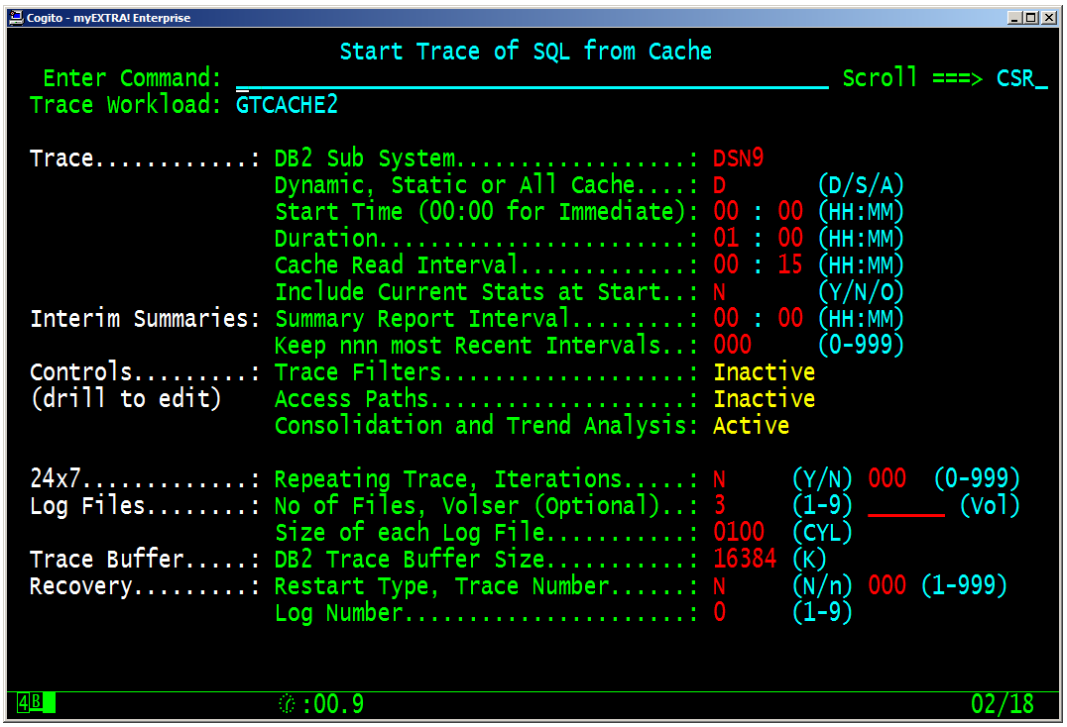


Figure 21 Set up Scheduled Cache Trace

Specify the required trace options exactly as if you were submitting the trace manually. See option 1.1 - Start Cache SQL Analysis for further details.

For example, if you want to run a five-minute trace every hour, set Trace Interval to 00:05.

After you have selected the required parameters press <**enter**> to generate the job stream.

You must edit the generated job stream to change *your.jobsched.hlq* to an appropriate JCLLIB where the generated job will be saved.

This job stream contains the following steps.

ALLOCDS	Allocate the required datasets for the workload
RUNVSAM3	Allocate and initialise VSAM report datasets
CPYJOB18	Copy Tracer Job to specified JCLLIB

- TYPE SUB in the command line to submit the job
- <**PF3**> to return to the EZ-Cache main menu.

*The generated job will create the required EZ-Cache job stream in a member in **JCLLIB** with a member name matching the current workload name.*

See also option 1.6 for further details of submitting the job using your automated job scheduling system.

1.6 Scheduler Job

Select this option (option 1.6) to generate the job stream that is to be submitted by your job scheduling system.

You must edit the generated job stream to change *your.jobsched.hlq* to the name of the JCLLIB where the Job stream was created for this workload using option 1.5.

Do not submit this job manually unless you want a trace to start immediately. This job is to be run by your scheduler. It will initiate the next instance of the EZDB2 regular Trace job every time it is run.

This job stream contains the following steps:

EDITJOB	EDIT the member in JCLLIB to create the next trace invocation for this workload.
CPYJOB18	Copy job to the internal reader to start the next trace.

Ensure the generated JCL points to the correct JCLLIB and save this job where it can be submitted from your job scheduler.

The member in JCLLIB (created using option 1.5) contains the job stream that will be tailored and submitted to the internal reader to start the next trace. You may wish to move this from JCLLIB to some other library where it will not be overwritten.

1.7 Set Up Started Task Cache SQL Analysis

It is possible to set up an EZ-Cache trace to be executed as a STARTED TASK.

Select option 1.7 to display the Start Cache trace panel as shown in the following figure:

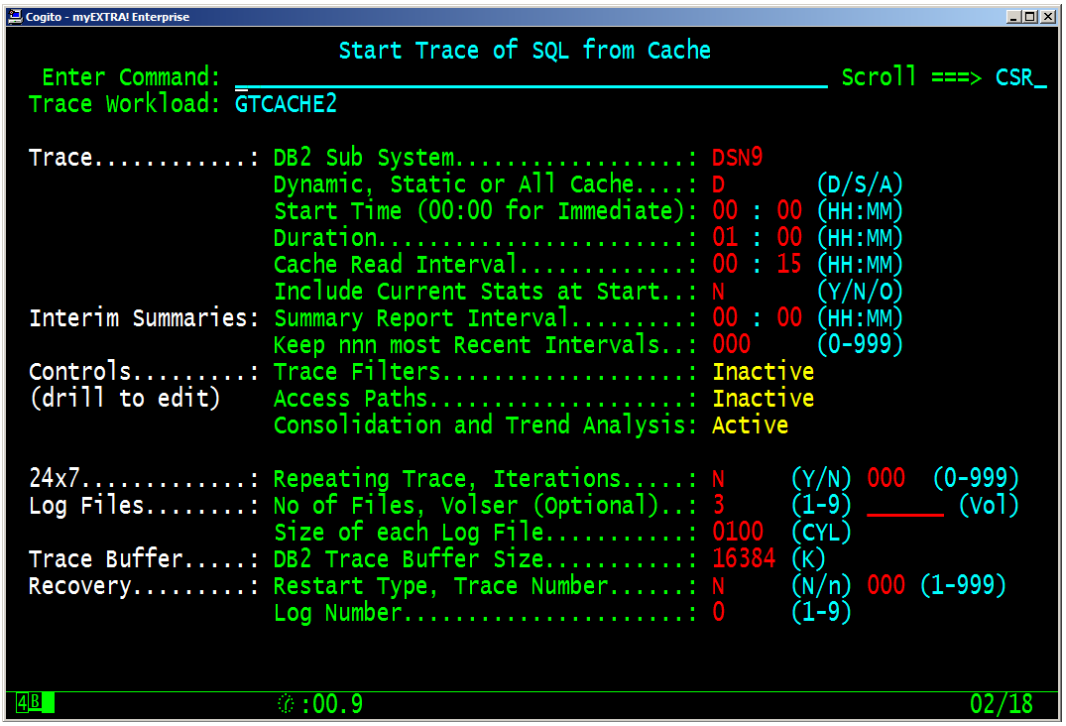


Figure 22 Set up Started Task Cache Trace

Specify the required trace options exactly as if you were submitting the trace manually. See option 1.1 - Start Cache SQL Analysis for further details.

After you have selected the required parameters press <**enter**> to generate the job stream.

The generated job stream must be edited as follows:

SET PROCLIB=your.proclib	Change *your.proclib* to the name of the JCL library that will contain the generated job stream, e.g. SYS1.PROCLIB
SET PROC=*wlname*	Change *wlname* to the name you want to use as the PROC name. The default name is the workload name.

This job stream contains the following steps.

ALLOCDS	Allocate the required datasets for the workload
RUNVSAM3	Allocate and initialise VSAM report datasets.
CPYJOB15	Convert Trace job to PROC and add it to the specified PROCLIB

- TYPE SUB in the command line to submit the job
- <**PF3**> to return to the EZ-Cache main menu.

The saved job stream may now be executed as started task in the normal way, for example using the SDSF **/S** command or using a job scheduler.

Each time the job is started it will automatically increment the trace number to initiate the next iteration of the trace.

2 EZ-Tracer

Select option 2 to display the EZ-Tracer Menu as shown in the following figure:

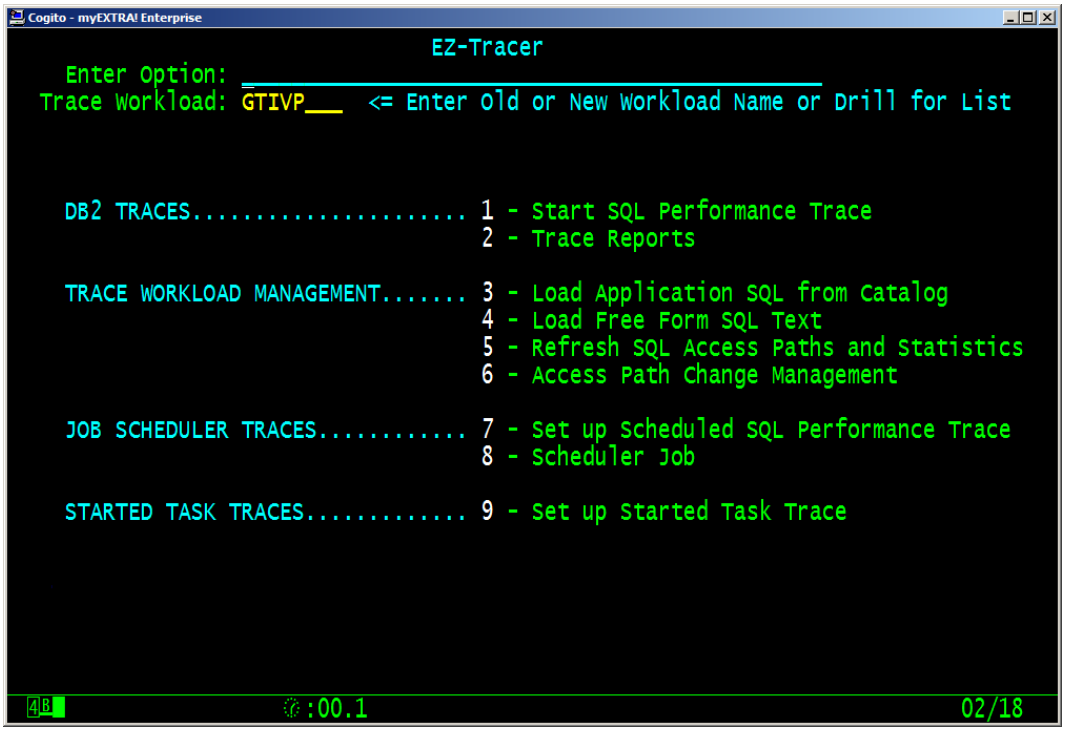


Figure 23 EZ-Tracer Main menu

The following options are displayed on this menu:

DB2 Traces

1.	**Start SQL Performance Trace**	Set the Trace Options and Start the Trace.
2.	**Trace Reports**	Display the Trace reports menu.

TRACE WORKLOAD MANAGEMENT

3.	**Load Application SQL from Catalog**	Supplement the Trace Workload with Static SQL loaded from the DB2 catalog for selected applications. You can optionally set the trace options to only trace executions of this pre-loaded SQL.
4.	**Load Free Form SQL Text**	Supplement the Trace Workload by adding Free Form SQL Text and Explain it to create Access Paths.
5.	**Load SQL Access Paths**	Generate the access paths for SQL if not generated at trace time, or refresh the access paths that were collected at trace time.
6.	**Access Path Change Management**	Switch to the EZ-DB2 Impact Analyzer for the current trace workload.

JOB SCHEDULER TRACES

7.	**Set up Scheduled Cache SQL Analysis**	This option is used to set up an EZ-Cache trace to be scheduled for automatic submission by a job scheduling system. Trace options are specified exactly as if submitting a manual trace.
8.	**Scheduler Job**	

		This option is used to display the Job to be submitted by the Job Scheduler. It is recommended this JCL be stored in a PDS used exclusively by Job Scheduler.

STARTED TASK TRACES

9.	**Set up Started Task Trace**	This option is used to set up an EZ-Tracer Trace to be executed as a STARTED TASK

These options are described in detail in the following sections.

2.1 Start SQL Performance Trace

The Start SQL Performance Trace option allows you to perform a trace to collect SQL trace data from the specified DB2 subsystem. The trace runs as a standalone batch job utility utilizing the standard DB2 instrumentation facility. Before starting the Trace process, you must set the Trace Options as appropriate for the current Workload

Execution of the Trace job includes the following job steps:

- Collecting Trace data
- Summarizing and Consolidating the trace data
- Extracting System Catalog statistics from the system catalog for the applicable databases
- Generating the reports

Further information about the job steps can be found in Job Steps Summary and the Appendix of this User Guide.

Specifying Trace Options

Select option 2.1 to display the Start SQL Performance Trace panel as shown in the following figure:

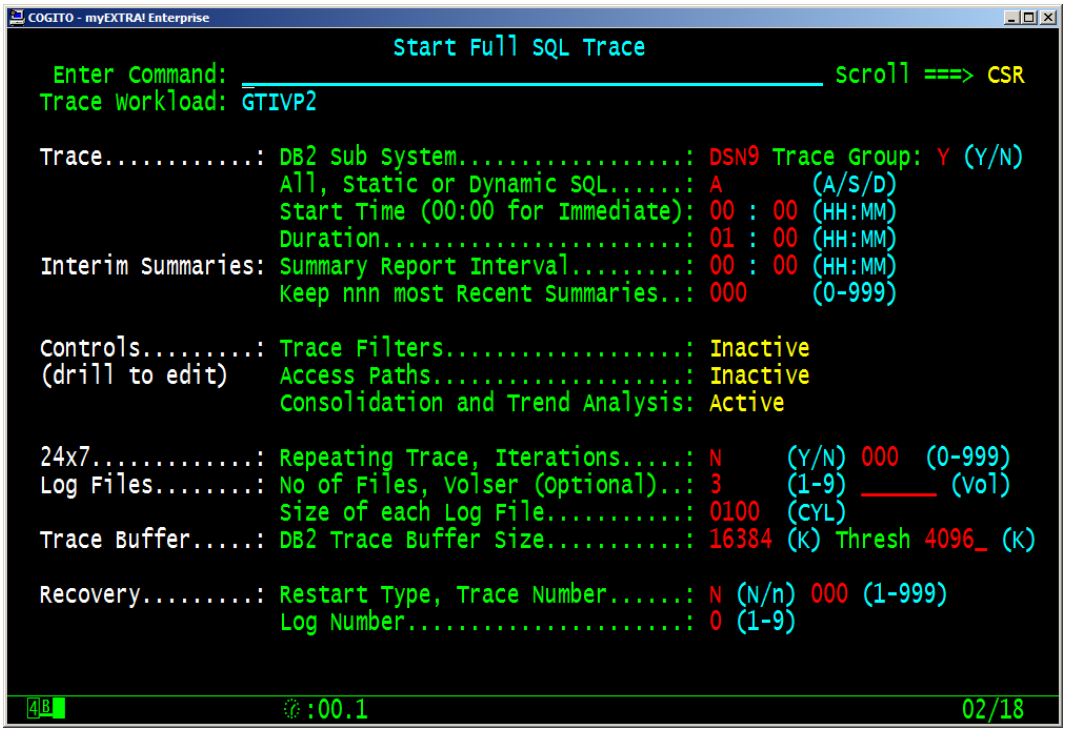


Figure 24 Start SQL Performance Trace

Most of the options are identical to those used in option 1.1 - Start Cache SQL Analysis. Refer to Section 1.1 for a description of the options not described here.

The Trace parameter settings are set to the default values shown below when a user creates their first workload. The actual settings used for each workload are remembered. However, when a new workload is created the initial settings are the default values.

If you usually use settings which differ from the defaults as standard for most new workloads you may wish to set these as the default initial settings for your new workloads. You can do this by issuing the command **SET PROFILE** when in the command line. This will save the current settings shown on the panel as the default initial settings for your userid and these will be used as the initial settings whenever you create a new workload. You can also use the **COPY WORKLOAD** command to copy the parameters from an existing workload. Refer to the **EZ-DB2 Commands Reference** for further information.

All, Static or Dynamic SQL

This parameter may have one of the following values:

A — Start a Performance Trace capturing both Static and Dynamic SQL.
This is the default value.

S — Start a Performance Trace capturing only Static SQL.

S might be used whenever a user wants to trace Dynamic Statement Cache and, at the same time, trace Static SQL though a regular trace. When running a regular trace in parallel with a Dynamic Statement Cache trace, the user must specify S in the regular trace otherwise all the Dynamic SQL will be double accounted.

D — Start a Performance Trace capturing only Dynamic SQL.

Thresh

The Buffer Threshold determines when EZ-Tracer is posted by DB2 to say that there is trace data to be processed. The default threshold is 25% of the Buffer Size, which allows time for EZ-Tracer to process the data before the Buffer fills up.

Once you have completed entry of the required parameters, press <**enter**> to validate. You are then prompted to press <**enter**> again to continue or **<PF12**> to cancel.

If you continue, the following pop-up panel is then displayed:

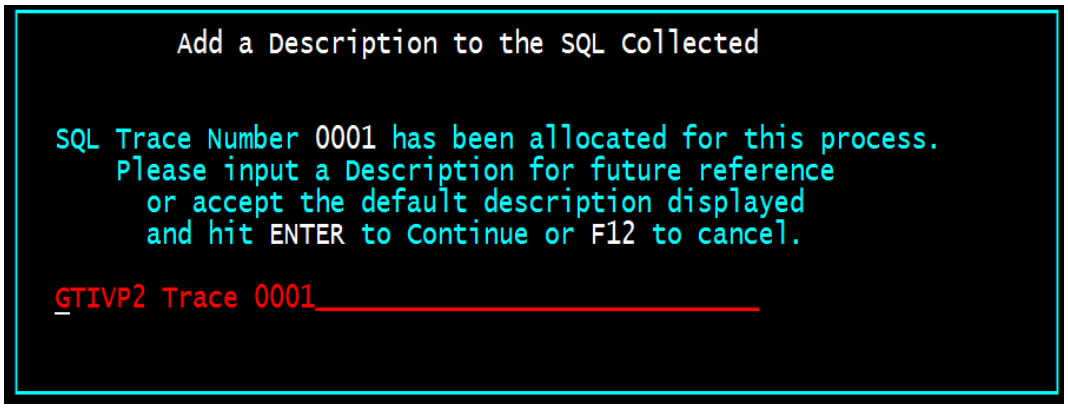


Figure 25 Add an SQL Trace Description

The panel displays the number of the next available SQL Trace File that will be created to contain the SQL collected for this Trace Workload.

You are prompted to enter a Description for this trace file. Enter a Description and press <**enter**> to display the generated job stream to start the DB2 trace.

- Type SUBMIT in the command line to submit the job for execution.
- Press the **<PF3>** key to return to the Start SQL Trace menu.

Trace Filter Parameters

The Trace Filter Parameters panel is displayed by DRILLING on the Active/Inactive indicator for Trace Filters Controls on the Start SQL Performance Trace panel.

The Trace Filter Parameters are displayed on 2 panels and you can use PF8/PF7 to page back and forth between the parameter screens.

The initial panel displayed is shown in the following figure:

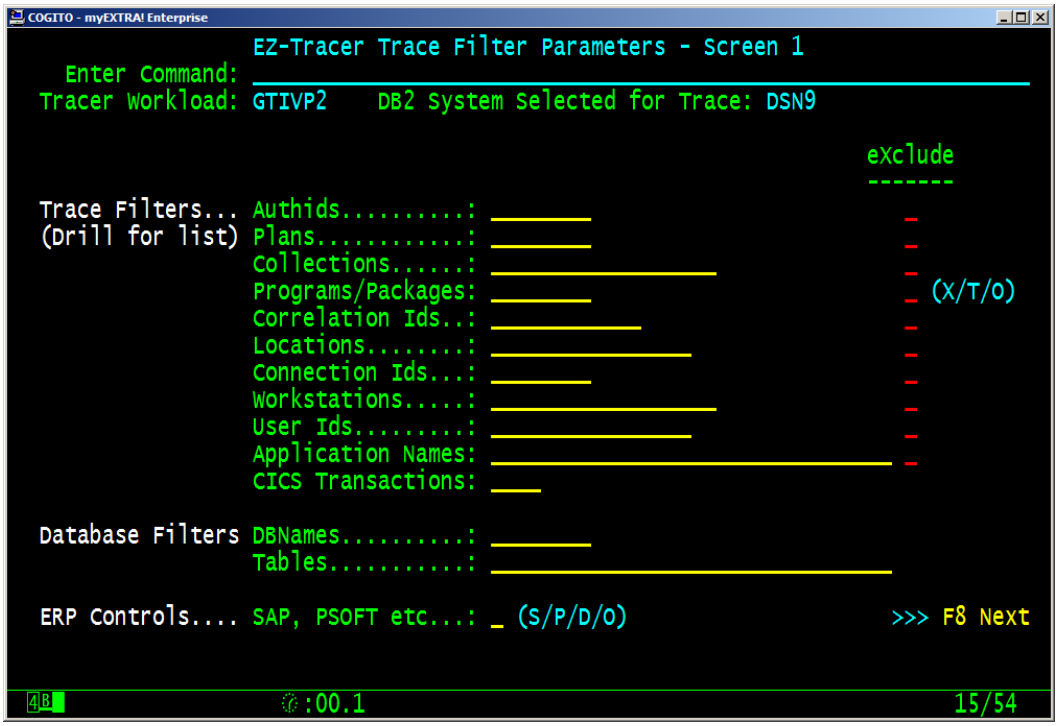


Figure 26 Trace Filter Parameters - Screen 1

Press PF8 will display the second Trace Filter Parameter panel:

Figure 27 Trace Filter Parameters - Screen 2

The following parameters may be specified on these panels:

Trace Workload — The name of the current Trace Workload.

DB2 System Selected for Trace — The DB2 subsystem ID of the DB2 system to be traced.

TRACE FILTERS

Specifying partial trace filters
Prior to V9, DB2 did not allow partial filters (wild-cards) but you may still enter partial filters and EZTracer will manually filter the trace data.
DB2 9 and higher. Partial filters are supported by the DB2 Trace command.
*To use partial filters specify '_' in any position and/or terminating with * to define a prefix e.g.*
ABC for anything starting ABC.*
A_C for anything starting with A in position 1 and C in position 3.*

Authids — Constrain the trace to specific AUTHID names for which trace information will be collected. The AUTHIDs specified must be primary AUTHIDs.

Enter **X** in the eXclude field to exclude the specified AUTHIDS.

To enter more than one AUTHID, DRILL <PF4> on the field to display a list. You can specify up to eight AUTHIDs. If you specify more than one AUTHID, you can only use one value for PLAN and LOCATION.

Plans

Constrain the trace to specific PLAN names for which trace information will be collected.

Enter **X** in the eXclude field to exclude the specified PLANs.

To enter more than one PLAN, DRILL <PF4> on the field to display a list. You can specify up to eight PLANs. If you specify more than one PLAN, you can only use one value for AUTHID and LOCATION.

Collections

Constrain the trace to specific COLLECTION names for which trace information will be collected.

Enter **X** in the eXclude field to exclude the specified COLLECTIONS.

To enter more than one Collection, DRILL <PF4> on the field to display a list.

You may enter up to 4 collection names.

In DB2 9 or higher, DB2 will accept a single PKGCOL or XPKGCOL in the Trace command. Specifying more than one filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

Programs/Packages

Constrain the trace to specific DBRM or Program names for which trace information will be collected.

Following each PROGRAM entry, there is a 1-character field in which you can enter a control character, as follows:

X These programs are to be excluded from the trace.

T Means the SQL for this program and all other programs in the same Thread are to be traced and the trace is to be stopped when this program terminates. If T is specified it MUST be the first and only program in the list.

O Means the SQL for this one program ONLY is to be traced and the trace is to be stopped when this program terminates. If O is specified it MUST be the first and only program in the list.

To enter more than one Program, DRILL <PF4> on the field to display a list. You can specify up to eight program names.

In DB2 9 or higher, DB2 will accept a single PKGPROG or XPKGPROG entry in the Trace command. Specifying more than one filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

Correlation IDs

Constrain the trace to specific Correlation ID for which trace information will be collected

Enter **X** in the eXclude field to exclude the specified CORRIDs.

To enter more than one Correlation ID, DRILL <PF4> on the field to display a list. You may enter up to 4 Correlation IDs.

In DB2 9 or higher, DB2 will accept a single CORRID or XCORRID in the Trace command. Specifying more than one filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

Locations

Constrain the trace to specific LOCATIONs for which trace information will be collected.

Enter **X** in the eXclude field to exclude the specified LOCATIONS.

To enter more than one LOCATION name, DRILL <PF4> on the field to display a list. You may enter up to 8 LOCATIONS.

In DB2 9 or higher, DB2 will accept a single LOCATION or XLOC in the Trace command. Specifying more than one filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

Connection IDs

Constrain the trace to specific Connection IDs for which trace information will be collected.

Enter **X** in the eXclude field to exclude the specified Connections.

To enter more than one Connection ID, DRILL <PF4> on the field to display a list. You may enter up to 8 Connection IDs.

In DB2 9 or higher, DB2 will accept a single CONNID or XCONNID in the Trace command. Specifying more than one

filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

Workstation

Constrain the trace to specific Workstation names for which trace information will be collected.

Enter **X** in the eXclude field to exclude the specified Workstation names.

To enter more than one Workstation, DRILL <PF4> on the field to display a list. You may enter up to 8 Workstation names.

In DB2 9 or higher, DB2 will accept a single WRKSTN or XWRKSTN in the Trace command. Specifying more than one filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

User ID

Constrain the trace to specific USER IDs for which trace information will be collected.

Enter **X** in the eXclude field to exclude the specified USER IDs.

To enter more than one USER ID, DRILL <PF4> on the field to display a list. You may enter up to 8 USER IDs.

In DB2 9 or higher, DB2 will accept a single USERID or XUSERID in the Trace command. Specifying more than one filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

Application Name

Constrain the trace to specific Application names for which trace information will be collected.

Enter **X** in the eXclude field to exclude the specified Application names.

To enter more than one Application name, DRILL <PF4> on the field to display a list. You may enter up to 8 Application names.

In DB2 9 or higher, DB2 will accept a single APPNAME or XAPPNAME in the Trace command. Specifying more than one filter (or < DB2 9) will result in EZ-DB2 tracing all data and filtering manually.

CICS Transactions

Constrain the trace to specific CICS transaction names for which trace information will be collected.

To enter more than one CICS transaction name, DRILL <PF4> on the field to display a list. You may enter up to 8 CICS Transaction names.

Note: Filtering by CICS transaction name is not supported by the DB2 Trace command. Specifying this filter will result in EZ-DB2 tracing all data and filtering manually.

Database Filters

DBNames

Filter the Trace to only collect information for the specified DBNAMES. You may specify up 1-8 DBNAMES in the Trace Filter parameters.

Drilling on database name shows a list of all the available databases for the subsystem. You can select one or more. If you have selected a single database, and then drill on table name, it will display a list of valid table names for that database. You may also enter a DBName terminating with '*' to display a list of DBNAMES beginning with that prefix.

e.g.: Specify ABC* for a list of all DBNAMES with prefix ABC.

Note when using DBNAME or TABLE name filters all data will be deleted from the Workload that does not match the filters specified. If you do not wish to lose any existing data then clear these filters and use a new workload instead.

Tables

Enter a TABLE name by which you want to limit the trace, or a prefix and * to trace all tables with that prefix.

e.g.: ABC to trace ABC only or ABC* for all tables starting ABC.

You may enter more than a single table name only if you have selected a single database name. To enter more than one Table, DRILL <PF4> on the field to display a list of valid table names for the database name selected.

ERP Controls

SAP, PSoft etc. (S/P/D/O)

Certain applications, typically SAP®, PeopleSoft and Other JDBC type applications make use of the SET CURRENT SCHEMA as a way of specifying the object qualifiers to be used by an application. This has the benefit of making the application code transportable. However, this presents a challenge for EZ-Tracer/Cache as the SET CURRENT SCHEMA may have been issued before the trace started, and hence Tracer/Cache has no way of knowing how to properly qualify the subsequent SQL.

The default action is for EZ-Tracer/Cache to use the current AUTHID as the qualifier; however, this may not be correct.

To resolve this issue, you may now specify the desired mapping of AUTHID to Object Qualifier.

On the Trace Filters panel the ERP controls option can now be used to specify the qualifier translation.

The following values may be specified:

S This is an SAP/R3® application.

P This is a PeopleSoft application

O This is some Other ERP application.

After entering the applicable value, DRILL on the field to display the AUTHID/Qualifier translation panel as shown in the following figure:

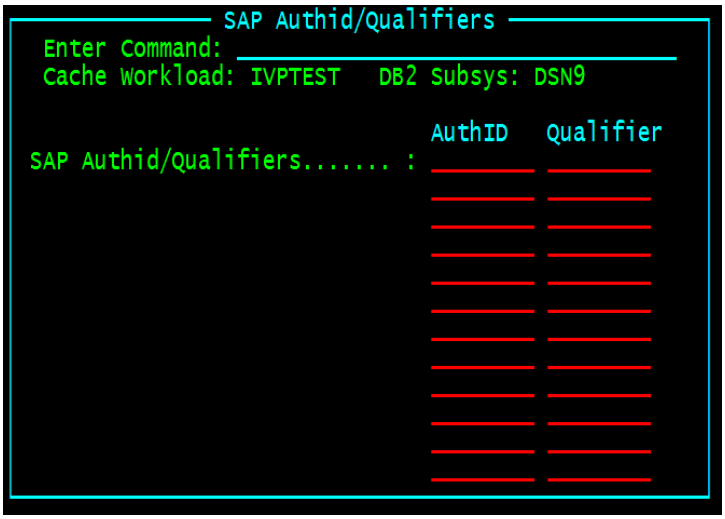


Figure 28 SAP Authid/Qualifiers

In this panel, you can enter up to 12 values for AUTHID and the associated default QUALFIER to be used.

Additional processing is required when copying the Catalog statistics for an SAP® workload to the analysis environment. Because of the dynamic nature of the SAP® environment, there is no guarantee that a table will be in the same Tablespace in the analysis environment as the live environment, and even corresponding Indexes may not have the same names in the Live and analysis environments. For further information, refer to the EZ-Stats User Guide. Appendix - Processing SAP® Workloads

It is also necessary to trace the dynamic statement cache rather than using the standard IFCID trace, as there is no guarantee that Tracer would collect the necessary PREPARE information required to capture the dynamic statement text.

It is not possible to mix SAP® and non-SAP® traces in the same Trace Workload. If you wish to analyze SAP® and non-SAP® workloads, you must create separate Trace workloads

D Enter D if you are using this workload to load SQL from CA Detector ®. Refer to Appendix Importing CA Detector Data for more information.

TRACE Sampling

Sample Percentage

See also Sample Interval.

You may optionally elect to reduce the overhead of long running traces by sampling data instead of collecting all data.

Set Sample Percentage to the sample value required e.g. 25% and a value in the Sample Interval (the default is 10 minutes, Summary Interval or Trace Duration, whichever is the least). The trace will then execute for 25% of sample interval, halt until the interval is complete and then resume for another 25% of sample interval etc.

Each Sample should be at least one minute to maximise chances of tying up OPEN/FETCH/CLOSE with Cursor Operations.

It should only be necessary to use this feature when running long traces against a DB2 Subsystem on an overloaded CPU.

Sample percentage may be specified in the range 00-90%.

00	Disables this feature. ALL activity will be traced.
01-90	Any numeric in this range will cause tracer to be active for this percentage of the Sample Interval. For example, if Sample Percentage is set to 50 and Sample Interval set to 10, then Tracer will be active for 5 minutes and wait for 5 minutes.

The default value is 00, i.e. Sampling is disabled.

Sample Interval

See also Sample Percentage. Used in conjunction with Sample Percentage if trace sampling is required rather than a full trace.

Specify the interval in the range 0-60 minutes. The trace will be active for a percentage of the Sample Interval, as defined by Sample Percentage.

The default is 10 minutes, Summary Interval or Trace Duration, whichever is the least. Each Sample should be at least one minute to maximise chances of tying up OPEN/FETCH/CLOSE with Cursor Operations

SQL TRACE THRESHOLDS

*The SQL Trace Thresholds specified here are not to be confused with the Minimum CPU MS, Elapsed MS and Getpages specified in **Trace Each SQL Execution.**. **Trace Each SQL Execution** thresholds relate to the capture of individual SQL execution data with Literal and HV values. The SQL Trace Thresholds specified here relate to the normal consolidated SQL data capture.*

Minimum CPU MS

Limit the trace to only record SQL that exceed a CPU threshold. The value specified is in milliseconds.

Minimum Elapsed MS

Limit the trace to only record SQL that exceed an elapsed time threshold. The value specified is in milliseconds.

Minimum Getpages

Limit the trace to only record SQL that exceed a specified number of GetPages.

Include SQL with Errors

This option allows you to capture SQL with errors (negative SQLCODE) even for SQL that do not trigger any SQL Trace thresholds.

The parameter can have one of the following values:

Y - Capture SQL execution instances including literal and host variable values for SQL with Errors even if the SQL does not trigger one of the SQL execution thresholds.

N - Do not capture SQL execution instances for SQL with errors unless the SQL triggers one of the execution thresholds.

Aggregate Fetches

If Aggregate Fetches is Y then the threshold is applied to the entire cost of each Cursor operation, i.e. the total of OPEN and all Fetches.

TRACE Each SQL Execution

*The **Trace Each SQL Execution** Thresholds specified here are not to be confused with the Minimum CPU MS, Elapsed MS and Getpages specified in **SQL Trace Thresholds** specified previously. **Trace Each SQL Execution** thresholds relate to the capture of individual SQL execution data with Literal and HV values. The SQL Trace Thresholds specified previously relate to the normal consolidated SQL data capture.*

Minimum CPU MS

Collect SQL execution data for SQL that exceeds this Minimum CPU threshold.

You may collect Host Variable Values and Dynamic SQL Literal Values for SQL that exceed a threshold of CPU specified in milliseconds. You can then view all executions of SQL that exceed the threshold seeing the Host Variables and literals for those SQL which generate the highest cost.

Note that taking this option will generate a LOT more trace data so we recommend the Thresholds be set high so only exceptional SQL costs get recorded at this level of detail.

Minimum Elapsed MS

Collect SQL execution data for SQL that exceeds this Minimum Elapsed time threshold.

You may collect Host Variable Values and Dynamic SQL Literal Values for SQL that exceed a threshold of Elapsed time specified in milliseconds. You can then view all executions of SQL that exceed the threshold seeing the Host Variables and literals for those SQL which generate the highest cost.

Note that taking this option will generate a LOT more trace data so we recommend the Thresholds be set high so only exceptional SQL costs get recorded at this level of detail.

Minimum GetPages

Collect SQL execution data for SQL that exceeds this Minimum Getpage threshold.

You may collect Host Variable Values and Dynamic SQL Literal Values for SQL that exceed a threshold of Getpage requests. You can then view all executions of SQL that exceed the threshold seeing the Host Variables and literals for those SQL which generate the highest cost.

Note that taking this option will generate a LOT more trace data so we recommend the Thresholds be set high so only exceptional SQL costs get recorded at this level of detail.

Record Host Variable Values

Specifies whether the trace will collect individual SQL executions with Host Variable values.

Note: Because of potential volume of data that may be generated, it is recommended that you only collect SQL execution data for SQL that exceeds a minimum threshold value. See Minimum CPU, Elapsed and GetPages.

Y - Collect SQL executions with Host Variable Values

N - Do not SQL executions

The default value is N.

Record Literal Values

Specifies whether the trace will collect individual SQL executions with Literal values.

Note: Because of potential volume of data that may be generated, it is recommended that you only collect SQL execution data for SQL that exceeds a minimum threshold value. See Minimum CPU, Elapsed and GetPages.

Y - Collect SQL executions with Literal Values

N - Do not SQL executions

The default value is N.

Record SQL with Neither

Specifies whether the trace will collect individual SQL executions with neither Host Variable nor Literal values.

Note: Because of potential volume of data that may be generated, it is recommended that you only collect SQL execution data for SQL that exceeds a minimum threshold value. See Minimum CPU, Elapsed and GetPages.

Y - Collect SQL executions without Host Variable and Literal values.

N - Do not collect SQL Executions.

The default value is Y.

TRACE CONTROLS

Max Kept Traces

If you wish to keep only a certain number of traces, then enter a value for the number of traces to be retained. A value of 0 means all traces will be kept, otherwise the oldest traces will be deleted from the workload as new traces are generated with the nn most recent retained. Note that the workload totals and reports will still report values for all traces executed

Deleting old traces frees up DASD and you lose the ability to drill into the deleted traces.

Trace Pre-Loaded SQL Only

Enter YES if you want to trace just those SQL already loaded into the Trace Workload.
Note: This is generally only applicable if you have seeded the Trace Workload with Application SQL loaded from the Catalog. The Workload may also be seeded with SQL using the Load Free Form SQL option.

This method is very useful if you want to be able to generate reports for a specific group of Plans or Packages. All other SQL in the Trace will be discarded.

*Another possible use. You could use the Load Free Form SQL text to load into the workload an example of a dynamic SQL that you wish to trace. Then you could run a trace with **consolidation** turned off, and **Trace Pre-loaded SQL Only** set to Y. EZ-Tracer will capture all occurrences and variations of that SQL, including SQL with different literal values, and report these all as different SQL instances. Refer to Load Free Form SQL option 2.4 for further information.*

The default value is NO.

Refer to section Load Application SQL from Catalog2.3 or details of Load Application SQL from the catalog.

Refer to section 2.4 for details of Load Free Form SQL Text.

Activate User Exit

(Y/N)

A user exit can be activated that allows user manipulation of the trace data before being written to the SQL log file for consolidation processing.

A sample user exit (EZTREXT1) is provided in the installation SRCLIB that modifies the AUTHID based upon a user supplied conversion table.

For example,

AC% is converted to AC_GRP,
DC% is converted to DC_GRP
etc.

In this way, all possible AUTHIDS may be consolidated to a few departmental groups in a way that is meaningful to the user.

Specify Y to activate the user exit.

For further information about EZ-EXTREXT1 refer to the Appendix.

Extract Catalog Stats

(Y/N)

Enter Y is you want the catalog statistics to be collected automatically for the databases reference by this workload at the end of the trace.

Note that this parameter works in conjunction with the **Refresh statistics** option on the access path controls. By default, the stats job that runs at the end of a tracer/cache run will only collect statistics for new databases that haven't previously been captured, unless you specify **Refresh statistics** = Y, in which case statistics for ALL databases will be collected.

If **Extract Catalog Stats** is set to N, then no statistics will be collected, whatever the value of **Refresh statistics**.

TRACE IFCIDs

You can suppress which IFCIDs are traced.

Fetches

Fetches are the highest volume SQL executed and generally represent 60-80% of the trace depending on filter parameters set. To minimize the CPU cost of performing the trace you may elect not to collect statistics relating to the execution of fetches for cursor operations. This will not affect the data collected for any other type of SQL.

Where a result set is generated at OPEN then the cost of fetches is trivial and cannot be tuned but the cost of OPEN may well be affected by poorly coded SQL or unsuitable indexes and this should be the focus of your analysis.

Where no result set is generated at OPEN then the cost is at FETCH then a full trace will be required for proper analysis.

The following parameter values may be specified:

Y - Trace FETCHES

N - Do not trace FETCHES

The default value is Y.

Selects

Y - Trace SELECTS

N - Do not trace SELECTS

The default value is Y.

Insert/Update/Delete

Y - Trace INSERTS/UPDATES/DELETES

N - Do not trace INSERTS/UPDATES/DELETES

The default value is Y.

Parallelism

Y - Trace PARALLELISM

N - Do not trace PARALLELISM

The default value is Y.

Prepares	**Y** - Trace Prepares **N** - Do not trace Prepares. The default value is Y.
Cursors	**Y** - Trace Cursors **N** - Do not trace Cursors The default value is Y.
Rid Failures	**Y** - Trace RID Failures **N** - Do not trace Failures The default value is Y.
CALLs	**Y** - Trace CALLS **N** - Do not trace CALLS The default value is Y.

When you have completed entry of the required parameters press **<PF3>** key to save the parameters and to return to the Start SQL Trace Panel.

Access Path Controls

The Access Path Controls panel is displayed by DRILLING on the Active/Inactive indicator for Access Path Controls on the Start SQL Performance Trace panel.

The following panel is displayed:

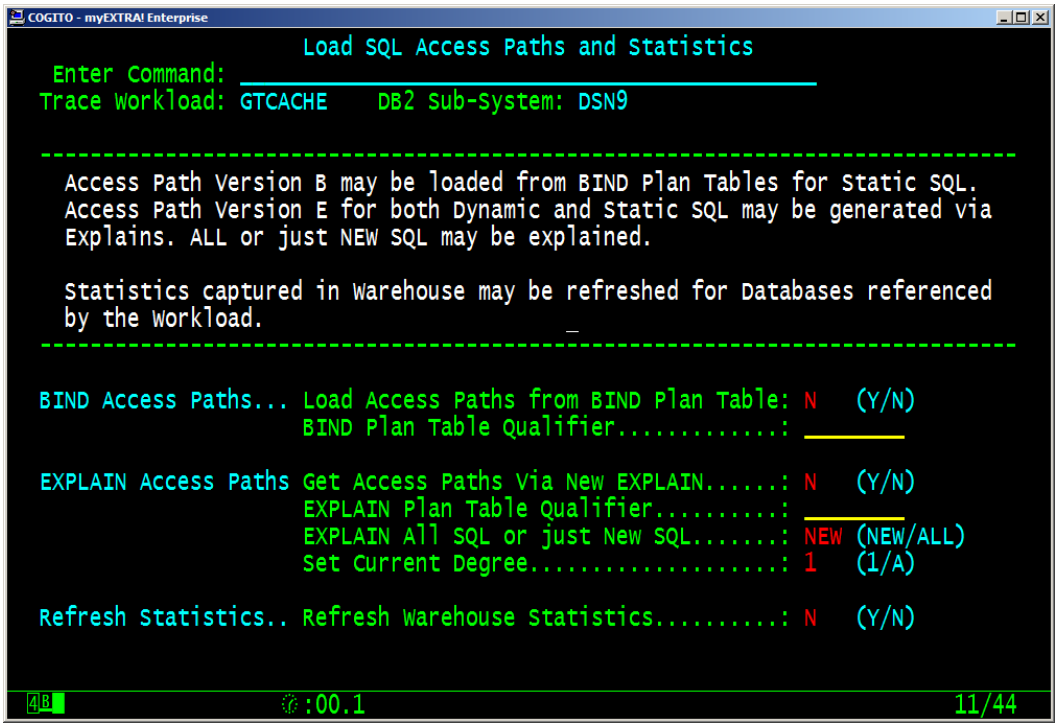


Figure 29 Specify Access Path Controls

Access Paths for SQL captured in the trace may be obtained by extracting from the applicable BIND PLAN Tables for STATIC SQL and /or explaining the SQL when the trace is terminated. By default, only NEW SQL where the Access Path is not already present will be explained, but you may opt for ALL SQL to be explained.

The BIND PLAN Table access path will generate Version B of the access Path data, The EXPLAIN process will generate Version E of the Access Path Data.

Note that if you do not choose to generate access paths when the trace terminates, you can subsequently add the access paths for the SQL in the trace using Options - 1.3 or 2.5 – Refresh access paths and Statistics.

The following parameters may be specified-

BIND Access Paths

Load Access Paths From BIND Plan Table

Specify whether you wish to get access paths from Plan table(s) used at BIND Time.

Y - Specify Y If you want to get the access paths from the Plan Table used at BIND time. You must have specified EXPLAIN(YES) when performing the BIND. This will load or refresh the current Version B access path for each SQL in the workload.

SQL not found in the BIND Plan Table will be Explained now (if you have also specified Get Access Paths via New Explain). This will load or refresh the current Version E access path for each SQL in the workload

N - Specify N if you do not want to get access paths from a BIND Plan table.

BIND Plan Table Qualifier

Specify the applicable qualifier(s) to be used to identify the BIND Plan table(s) to be used.

DRILL <PF4> be presented with a list of available PLAN_TABLEs from which you can select up to 15.

EXPLAIN Access Paths

Get Access Paths via New Explain

Specify whether you wish to get access paths by explaining the SQL.

Y - Specify Y to Explain the SQL using the specified PLAN table. This will load or refresh the current Version E access path for each SQL in the workload.

N - Specify N if you don't want to explain the SQL. This is the default value.

EXPLAIN Plan Table Qualifier

Specify the applicable qualifier to be used to identify the PLAN_TABLE to be used for the explain.

DRILL <PF4> to be presented with a list of available PLAN_TABLEs from which you can select the applicable qualifier

EXPLAIN All SQL or just New SQL

NEW - Only new SQL where the access path is not already present will be explained. This is the default value.

ALL - Explain all SQL.

Set Current Degree

A Set CURRENT DEGREE to ANY, prior to issuing the explains. This will allow parallelism, if parallelism is enabled for the subsystem.

1 Leave CURRENT DEGREE as default 1, to disable parallelism.

Refresh Statistics

Refresh Warehouse Statistics (Y/N)

At the end of each trace, statistics are captured to the statistics repository (warehouse) for each new database encountered. These are the statistics that are displayed when a user drills to a database object (e.g. table or index) from the access path display.

The access paths may be generated at the end of the trace, or explicitly using option EZ-Cache option 1.3 or EZ-Tracer option 2.5 to refresh the access paths.

By default, the statistics are captured for a given workload only the first time that the database is encountered. This means that it is possible for the statistics to become *stale*, and possibly not correspond to the statistics that were present at the time the access path was generated.

The Refresh Catalog statistics option in EZ-Cache and EZ-Tracer may be used to refresh the statistics each time the access paths are generated.

Y Specify Y to force all statistics to be refreshed for the workload at the end of the trace.

N The statistic will not be refreshed. They will only be extracted the first time the database is encountered for this workload/

Note the option to EXPLAIN all SQL or just NEW SQL. To ensure that all access paths for all SQL are in synch with the statistics, then you should EXPLAIN ALL SQL and refresh the statistics.

When you have completed entry of the required parameters press **<PF3>** key to save the parameters and to return to the Start SQL Performance Trace Panel.

Consolidation and Trend Analysis Controls

The Consolidation and Trend Analysis Controls panel is displayed by DRILLING on the Active/Inactive indicator for Consolidation and Trend Analysis Controls option on the Start SQL Performance Trace panel.

The panel is used to specify the Consolidation and/or the Trend Analysis data collection criteria if required.

The following panel is displayed:

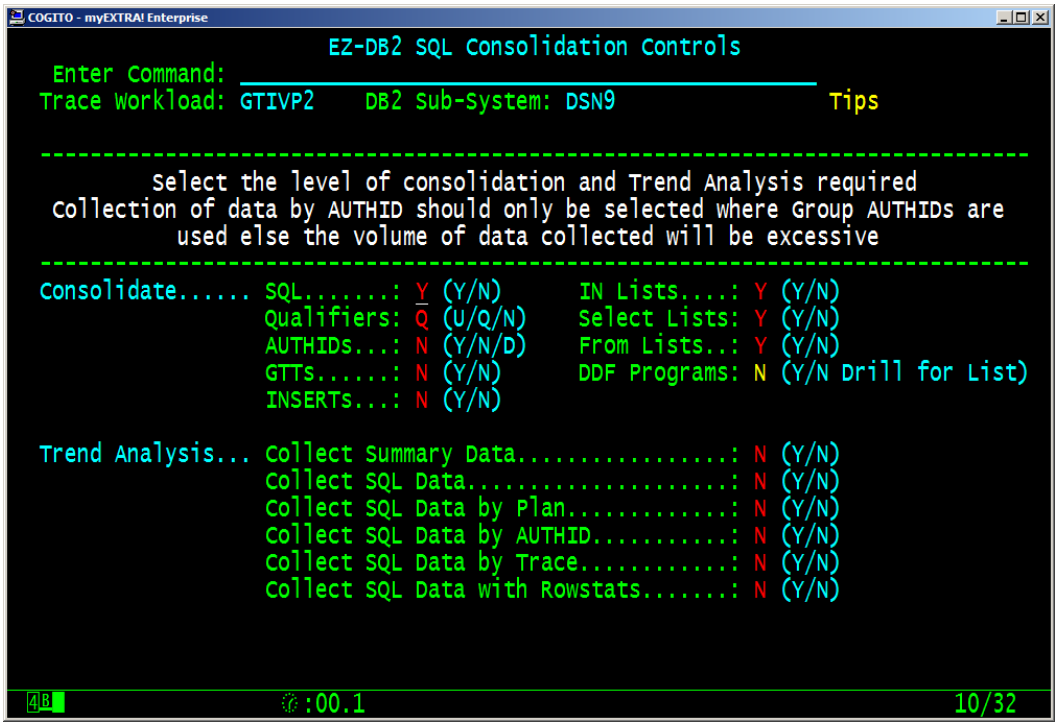


Figure 30 Specify Consolidation and Trend Analysis

For consolidation, the panel is used to specify whether SQL consolidation is active, and if so, to further specify the Consolidation actions to be taken.

A Trend Analysis database can optionally be created for each DB2 subsystem being traced. If it exists, EZ-Tracer will populate the database with relevant performance data at the end of each trace process. Refer to the EZ-DB2 Installation Reference Guide for further information regarding creation of the Trend Analysis Database.

The database is a standard DB2 database that can be accessed by user written programs. It is particularly useful for users wishing to perform trend analysis on other platforms. For example, Microsoft Windows7 users can use DB2 Connect to access the database directly from Excel7. The data available allows users to plot the performance characteristics of SQL by Plan, Program, Authid etc. over time, thereby making it easy to identify aberrations in the performance of particular statements, or to correlate variations in performance e.g., total/average CPU, Getpages etc. with particular events, such as application changes or systems maintenance such as Reorg or Runstats. The contents of the database

are documented in the EZ-DB2 Tables Reference Guide. This panel is used to define the data that is to be written to the Trend Analysis database.

The Consolidation and Trend Analysis Controls settings are set to the default values shown below when a user creates their first workload. The actual settings used for each workload are remembered. However, when a new workload is created the initial settings are the default values.

If you usually use settings which differ from the defaults as standard for most new workloads you may wish to set these as the default initial settings for your new workloads. You can do this by issuing the command SET PROFILE when in the Consolidation and Trend Analysis Controls panel. This will save the current settings shown on the panel as the default initial settings for your userid and these will be used as the initial settings whenever you create a new workload. Refer to the EZ-DB2 Commands Reference for further information about the SET PROFILE command.

The following parameters may be entered on this panel:

Consolidate

SQL

The following values may be entered for this parameter.

Y Consolidate Equivalent Dynamic SQL.

N Do not Consolidate Dynamic Equivalent SQL.

Dynamic SQL may appear many times in a Workload for the same Program and Statement number where the only difference in syntax is the value of an explicitly defined literal.

E.G. WHERE CUSTNUM = 12345 is equivalent to
WHERE CUSTNUM = 56789

Capturing all the distinct versions of the same SQL will result

in much larger Workloads and make it difficult to measure the true total and average costs of the SQL Statement as the costs will be spread over all the distinct versions. Consolidating these SQL into a single version will enable you to see the cost of all executions of the SQL in one. The SQL text captured will be the first version encountered.

The default value is Y

Qualifiers

U Enter U to consolidate UNQUALIFIED all equivalent SQL using different qualifiers for the same table names. This is the default and most effective form of consolidation which will consolidate equivalent SQL where the ALIAS, VIEW or TABLE qualifiers are not equal. For example: two SQL referencing ALIAS1.TableName and ALIAS2.TableName, where Tablename is the same, will be treated the same

and reported as one SQL. Likewise, the SQL for a package bound twice with different qualifiers.

Q Enter Q to consolidate QUALIFIED all equivalent SQL using different qualifiers for the same table names. For example: where a program is bound into two plans with different qualifiers then all the SQL will be reported as two different SQL, each with its own qualifier.

N Enter N to NOT perform any qualifier processing prior to consolidation. Two otherwise identical SQL, one coded fully qualified and one unqualified, will be reported as two different SQL.

The default value is N.

AUTHIDs

Y Enter Y to consolidate all SQL for AUTHIDs. There will be no breakdown of SQL Statistics by AUTHID.

D Enter D to accumulate trace data by AUTHID for Distributed Dynamic SQL only. Typically, GROUP AUTHIDs will be in use. SQL Statistics will be broken down by AUTHID for all DDF activity only.

N Enter N to accumulate trace data by AUTHID and allow full AUTHID reporting. Note This option should only be taken where GROUP AUTHIDs are extensively in use else the volume of data may be prohibitive.

The default value is D.

GTTs

Enter Y to enable consolidation of Global Temporary Table Names. In some applications, particularly Micro Strategy®, the same table may be defined many times using different table names. This results in many variations of the same SQL as each execution appears to reference different tables.

In other applications, the same table name may be used for what are in fact very different definitions. In either case enabling consolidation will result in the table definitions being analyzed and a unique internally generated name allocated to each unique table definition. The SQL that reference the tables are then updated to use the newly generated name. All equivalent SQL that reference each unique table will then be consolidated into one.

The default value is N.

INSERTS	Enter Y to enable consolidation of single row INSERT statements. INSERTs for the same table are consolidated when there are no SELECT sub-clauses regardless of how the column list is coded or defaulted. The default value is N.
IN LISTs	Enter Y to consolidate equivalent SQL with differing numbers of entries in IN LISTs. The default value is Y.
Select Lists	Enter Y to enable consolidation of columns in Select Lists. SELECT COL_A, COL_B and SELECT COL_B, COL_A are then treated as being the same SQL and consolidated into one. The default value is Y.
From Lists	Enter Y to enable consolidation of tables in From Lists. FROM TABLE_A, TABLE_B and FROM TABLE_B, TABLE_A are then treated as being the same SQL and consolidated into one. The default value is Y.
DDF Programs	Enter Y to consolidate SQL for specific DDF Program Names into the Program Name DDF. Typically, DDF program names are not meaningful as they are simply SQL Drivers so it does not help to see SQL execution frequencies and statistics for each version of the drivers. Consolidation reduces the volume of data collected and provides more meaningful reports. The default value is N. **Note**: You may DRILL on this field to see and update a list of DDF program names that will be consolidated as shown in the following figure:

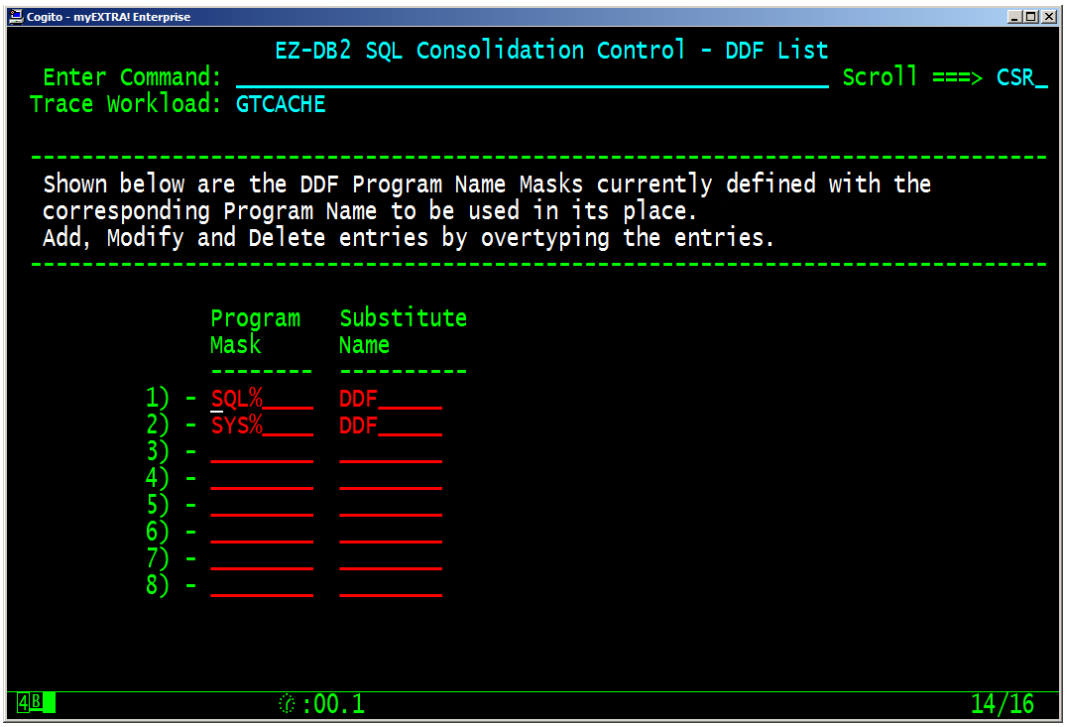


Figure 31 DDF Program consolidation list

Program name prefixes may be specified e.g. SYSN%.

Trend Analysis

Collect Summary Data

Enter Y to update the Trend Analysis database with Summary data for this trace.

The default value is N.

Collect SQL Data

Enter Y to update the Trend Analysis database with SQL data for this trace.

Enter D to delete the existing SQL Trace data for this Workload.

The default value is N.

Collect SQL Data by Plan

Enter Y to update the Trend Analysis database with SQL data by Plan for this trace.

Enter D to delete the existing SQL Trace data by Plan for this Workload.

The default value is N.

Collect SQL Data by Authid	Enter Y to update the Trend Analysis database with SQL data by Authid for this trace. Enter D to delete the existing SQL Trace data by Authid for this Workload. The default value is N.
Collect SQL Data by Trace	Enter Y to update the Trend Analysis database with SQL data by Trace for this trace. Enter D to delete the existing SQL Trace data by Trace for this Workload. The default value is N.
Collect SQL Data with Rowstats	Enter Y to update the Trend Analysis database with SQL data with Rowstats for this trace. Enter D to delete the existing SQL Trace data with Rowstats for this Workload. The default value is N.

Job Step Summary

The generated job contains the following internal job steps:

Job Step	Description
ALLOCDS	Allocate required datasets
ALOCRPTV[4]	Allocate VSAM Report datasets
RUNVSAM3[1]	Initialize VSAM Report datasets
ALOQSQLV	Allocate EZ-DB SQL VSAM Work Datasets
XOPTRC1	This is the main trace job that collects the SQL trace data from the applicable DB2 subsystem.
XOPTRC2	Analyze SQL Trace. This Step is submitted automatically at the end of the Trace Interval (or at the end of each Summary interval where Summary interval has been specified).
	The following job steps are executed after the final iteration of XOPTRC2
RUNPATH	Generate access path data if access path controls have been specified
SUMMARY	Accumulate Workload Summary. The data for the entire trace is made available to the Online Reporting panels.

[4]First time for this workload only

COMPRESS	Compress CATSTATS dataset in preparation for running RUNSTATD
RUNSTATD	Extract System Catalog statistics for Traced databases to the statistics Warehouse.
DELWORK	Delete EZDB2 trace VSAM datasets
DELLOGF	Delete EZDB2 trace Log datasets
REPSUB	Submit report jobs
DELGRPTS	Existing report files for the workload are deleted before they are recreated by the next step.
REPORTT	Generate Reports for Tracer - Current Trace
REPORTW	Generate Reports for Tracer - Workload Reports

For further details of these job steps refer to the Appendix.

Manually Stopping the Trace

It is possible to terminate the trace process at any time by entering the following modify command from within SDSF.

/F *jobname*,STOP or

/F *jobname*,SHUTDOWN

Where *jobname* is the job name as applicable for the trace job.

It is recommended that this command be used if you wish to terminate a trace before the expiration of the trace duration. When terminated using this command, the trace will end cleanly and the summarization and reporting functions will be executed for the currently active log file.

If you cancel the trace with a CANCEL or PURGE command, the data in the last logfile will be lost.

See also Trace Restart option in Start SQL Performance Trace.

Manually Extending the Trace

Occasionally, a user may wish to extend the duration of a trace that is already in progress.

To extend the trace time enter the command

/F *jobname*,ADD,*nnnn*

Where *jobname* is the jobname of the trace job and *nnnn* is the number of minutes to be added to the trace duration. Leading zeroes are mandatory, for example

/F CLGXB831,ADD,0010

to add 10 minutes to the trace time. *nnnn* should be in the range 1 – 1440 (24 hours).

The command may be entered any time after the trace has started and may be repeated if further extensions are required any time before the trace ends.

2.2 Trace Reports

The reports are common to SQL Warehouse Manager and both EZ-Cache *dynamic statement cache* traces and EZ-Tracer *Performance* traces.

*Depending on which EZ-DB2 component you are in, the report headers will say either **Trace Workload** or **Warehouse Workload**. When viewing Reports in the SQL Warehouse, it is not possible to DRILL down to individual traces, or trace summary intervals. There are also some minor differences in the available data displayed. For example, there is a more detailed breakdown of the DB2 Row statistics when viewing data collected using EZ-Tracer compared to EZ-Cache. This is due to limitations of DB2.*

Select option 2.2 to display the Reports menu. The following panel is displayed:

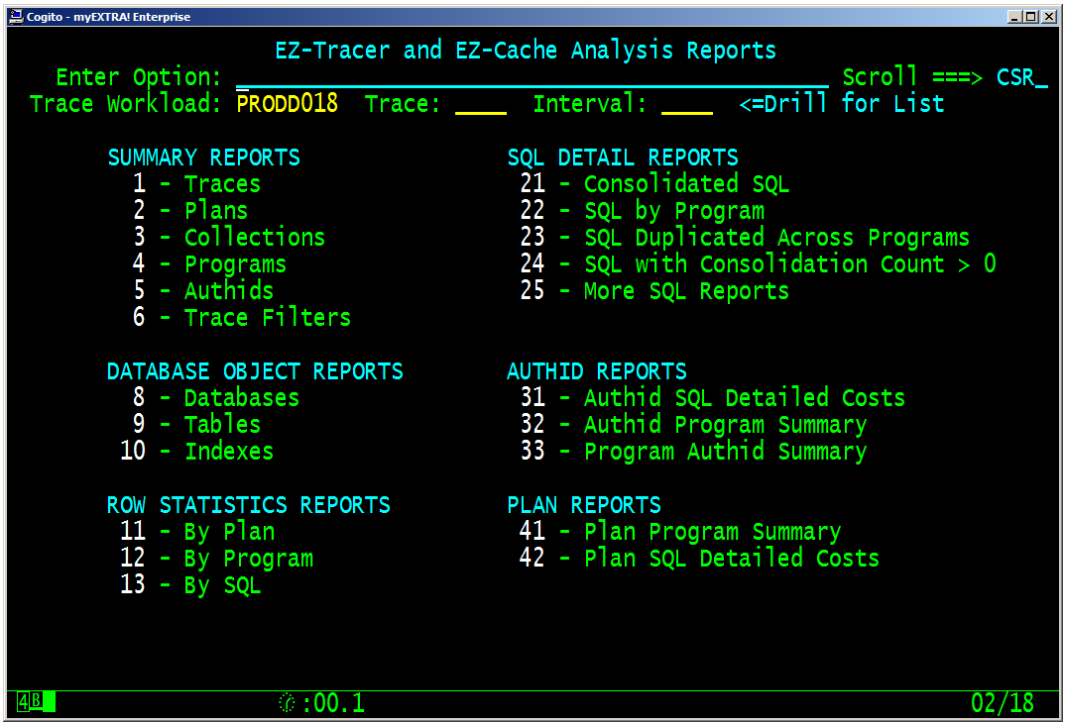


Figure 32 Trace reports menu

Refer to the EZ-DB2 Reports Guide for further information.

2.3 Load Application SQL from Catalog

You may use this option to supplement an existing Trace Workload with SQL loaded for specific applications from the System catalog.

You may also use this option to initially populate a Trace Workload with SQL for a specific application, and then use the Trace Filter parameters so that the trace only monitors the pre-loaded SQL and ignores any other trace activity. See Start Workload Trace option Trace Pre-Loaded Application SQL Only.

The SQL is loaded into the Workload with the Execution and Fetch Frequencies set to 1.

You may subsequently use the SQL Warehouse Manager function Refine SQL Workload Frequencies (option 6.5) to explicitly set the Execution and Fetch Frequencies for each SQL.

However, by pre-loading the SQL from the Catalog, and then using the Trace facility to monitor the pre-loaded SQL, you will collect actual SQL Frequency and Cost information for the application.

This combination of processing options allows you to not only analyse and optimise existing production applications, but also to model planned new applications or simulate predicted workload growth.

Note you can use the % sign in the same manner as a LIKE condition to load SQL from groups of PLANS or PACKAGES.

Select option 2.3 to display the Load Application SQL from Catalog panel as shown in the following figure:

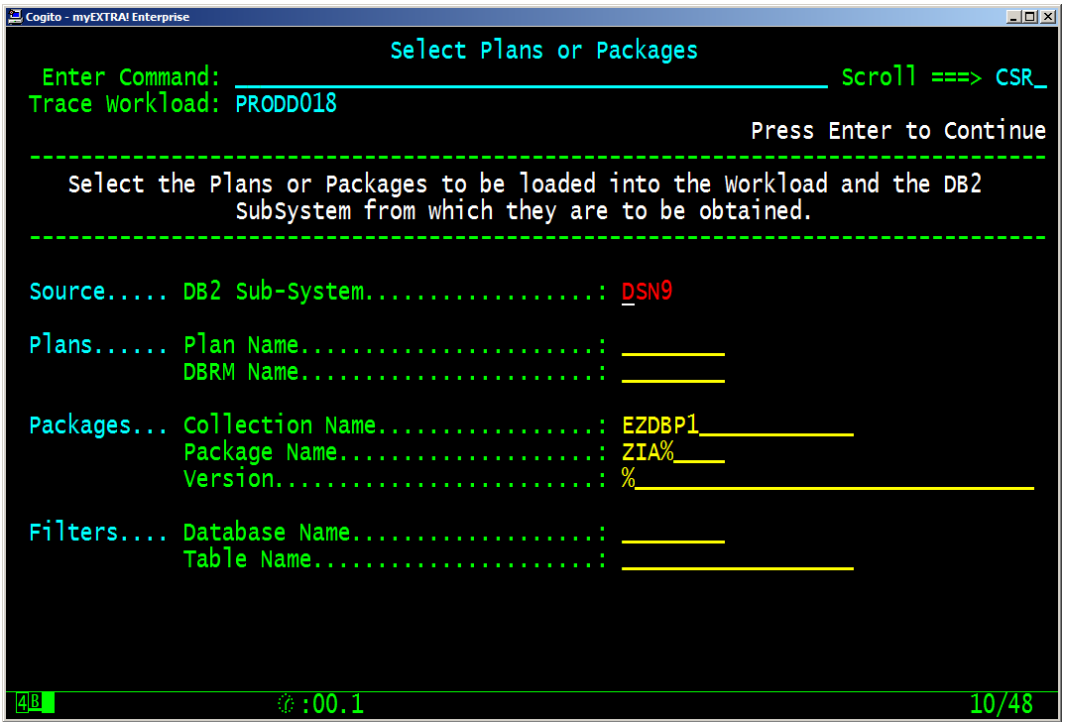


Figure 33 Load Application SQL From System Catalog

The following parameters may be specified on this panel.

DB2 Sub-System	Specify the DB2 subsystem ID of the system of the System Catalog you wish to load SQL from.
Plan Name	Specify the Plan Name and DBRM for the program that you want to extract the SQL from. **NOTE**: You can use the % sign in the same manner as a LIKE condition to load SQL from groups of PLANS or PACKAGES. **NOTE**: You can extract SQL from either a Plan or a Package but not both in the same run.
DBRM Name	Specify the Plan Name and DBRM for the program that you want to extract the SQL from. **NOTE**: You can extract SQL from either a Plan or a Package but not both in the same run.
Collection	Specify the Collid, Package name and optionally the Version for the program that you want to extract the SQL from. If Version is omitted, then Tracer will use the latest version of the Package in the catalog. **NOTE**: You can extract SQL from either a Plan or a Package but not both in the same run. **NOTE**: You can use the % sign in the same manner as a LIKE condition to load SQL from groups of PLANS or PACKAGES.
Package	See Collection.
Version	See Collection
Database Name	The SQL may optionally be filtered by Database Name to only capture SQL that reference the specified Database Name(s). This field will contain the database name when only a single database name has been selected. When more than one is selected, the field will display the word MULTIPLE. You may select up to 8 database names. However, if you also select Table Names only one database name may be selected. Use <PF4> to drill on the field to obtain a list of available Database names. Use a prefix and % to limit the displayed list to all databases with that prefix. E.g.: ABC% for all database names starting ABC.

EZ-DB2 will display a list of the database names available from the selected subsystem and will indicate any which have been selected.

If EZ-DB2 is unable to access the subsystem then a default panel will display showing any database names currently selected. You can manually enter database names using this panel.

Table Name

The SQL may also optionally be filtered by Table Name to only capture SQL that reference the specified Table Name(s).

This field will contain the Table name when only a single table name has been selected. When more than one is selected, the field will display the word MULTIPLE.

You may select up to 50 table names. However, when you select table names you must specify one and only one database name.

Use <PF4> to drill on the field to obtain a list of available Table names. Use a prefix and % to limit the displayed list to all tables with that prefix.

E.g. XYZ% for all table names starting XYZ.

EZ-DB2 will display a list of the table names available from the selected subsystem for the selected database name and will indicate any which have been selected.

If EZ-DB2 is unable to access the subsystem then a default panel will display showing any table names currently selected. You can manually enter table names using this panel.

If you have table names selected and you change the selected database name then all selected table names will be deselected.

Press **<enter>** to display the following pop-up panel:

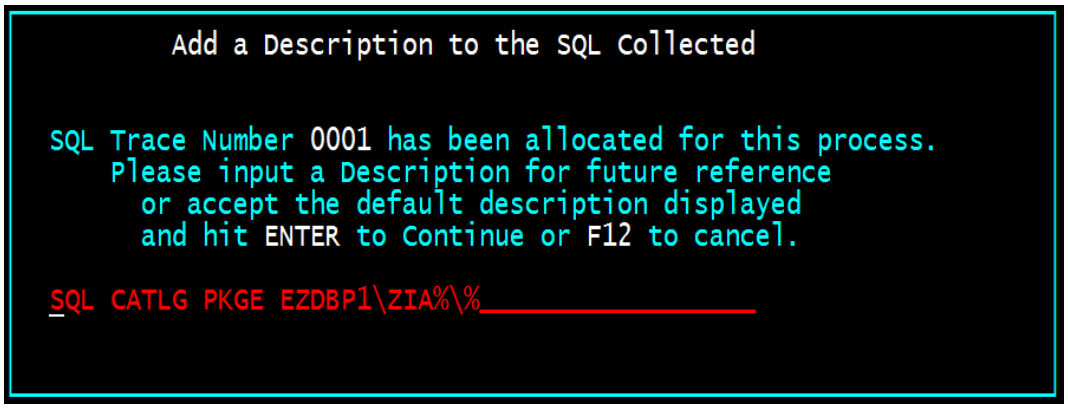


Figure 34 Add an SQL Input Description

The panel displays the number of the next available SQL Input File that will be created to contain the applicable SQL loaded from the catalog.

You are prompted to enter a Description for this trace file. Enter a Description and press <**enter**> to display the generated job stream to extract the SQL from the Catalog into the Trace File.

- Type SUBMIT in the command line to submit the job for execution.
- Press the **<PF3>** key to return to the EZ-Tracer menu.

Job Step Summary

The generated job contains the following internal job steps:

Job Step	Description
ALLOCDS	Allocate required datasets
ALOCRPTV[5]	Allocate VSAM Report datasets
RUNVSAM3[2]	Initialize VSAM Report datasets
ALOQSQLV	Allocate EZ-DB SQL VSAM Work Datasets

[5]First time for this workload only

RUNASQL	Load additional SQL from Plans and Packages into Trace file
XOPTRC2	Analyze SQL Trace.
SUMMARY	Accumulate Workload Summary. The data for the entire trace is made available to the Online Reporting panels.
RUNLOGO	Create SQLLOGO File. This is a consolidated SQLLOG file that is used when importing data into the SQL Warehouse.
COMPRESS	Compress CATSTATS dataset in preparation for running RUNSTATD
RUNSTATD	Extract System Catalog statistics for Traced databases to the statistics Warehouse.
DELGRPTS	Existing report files for the workload are deleted before they are recreated by the next step.
REPORTT	Generate Reports for Tracer - Current Trace
REPORTW	Generate Reports for Tracer - Workload Reports

For further details of these job steps refer to the Appendix.

2.4 Load Free Form SQL Text

This panel submits a job that will add additional SQL from a free-form SQL Text file to an existing Tracer Workload and Explain the SQL to generate Access Paths.

Put the SQL Text in a member in a PDS dataset and specify the PDS dataset name and member on the panel. When you specify a PDS dataset in the panel you can drill on the Member name field for a list of members to select from.

The SQL must be valid and be fully qualified or preceded by an appropriate SET CURRENT SQLID. You may specify multiple SQL in the same member but each SQL must be terminated by a semicolon. Comment lines (- - in cols 1and 2) may be specified for your own use.

You may optionally specify –PROGNAME *prog-name* to define the program name to be associated with the subsequent SQL statements. If No –PROGNAME parameter is specified, the program name defaults to SQLTEXT.

You should specify the DB2 Subsystem to be used for the Explain and the creator of the Explain Plan table to be used. You can drill on the Plan Table field for a list to select from.

Note: It is advisable to set the desired Consolidation Options from Trace Controls before submitting this job.

A major benefit of using this option is for Dynamic SQL when used in conjunction with the Trace Pre-Loaded SQL Only *option in the Trace Filter Specification Screen. Using this option you can load an example of a Dynamic SQL Statement into the Workload. All variations of this SQL will then be captured (i.e. where the literal values are different) and be reported as distinct versions of that SQL. These can then be explained, allowing Index usage and CPU changes to be reviewed from the generated reports. In addition the* Threshold Filter Settings *can also be used in conjunction with this option to just capture variations of a particular SQL where for example the CPU exceeds the specified threshold value. The free form SQL text which you load will be shown on reports with a program name of SQLTEXT. In addition, when viewing the reports you have the option of issuing a SORT HASH to sort the displayed report in descending sequence based on average CPU. Alternatively you can issue SORT HASH TOTAL_CPU to get the display in descending sequence based on TOTAL_CPU.*

Select option 2.4 to display the Load Free Form SQL Text panel as shown in the following figure:

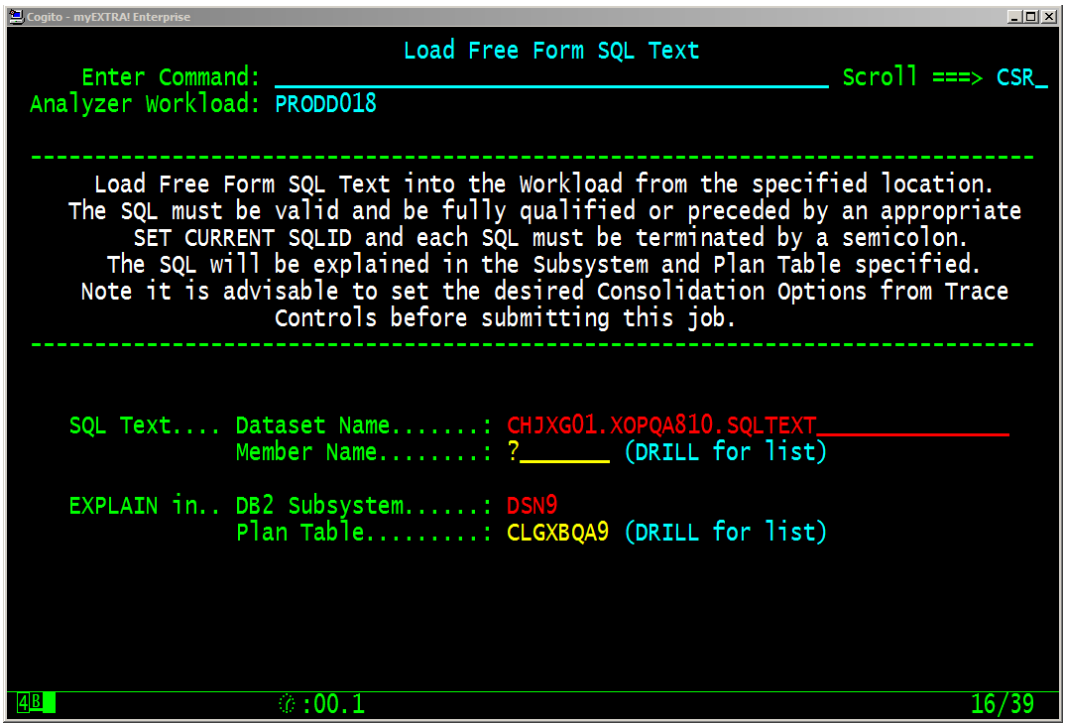


Figure 35 Load Free Form SQL text

The following parameters must be specified on this panel:

SQL Text

Dataset Name

Specify the name of the PDS Dataset which contains the SQL Text members.

Member Name

Specify the member of the PDS that contains the fully qualified SQL to be loaded.

Delimit each SQL statement with a semicolon or a comment line (-- in cols 1 and 2). Additional comment lines may be specified for your own use and will be ignored (except for the optional –PROGNAME parameter).

Enter a '?' to display a list of members in the applicable dataset from which you can make your selection.

You may select multiple members.

You can use the L(ocate) command to find the member name in the list.

EXPLAIN in

DB2 Subsystem

Enter the name of the DB2 subsystem in which the Explains are to be performed.

Plan Table	Specify the creator of the Plan Table into which the Explains are to be performed. Drill (PF4) to display a list of available plan tables.

Press **<enter>** to display the generated job stream to load the Free Form SQL.

- Type SUBMIT in the command line to submit the job for execution.
- Press the **<PF3>** key to return to the Tracer menu.

If you have received Explain failures the job will terminate with RC=4 and you should examine the log file for further information about the statements that have not been explained successfully.

Job Step Summary

The generated job contains the following internal job steps:

Job Step	Description
XOPALOC	Allocate Datasets (If required)
DELSQLV	Delete EZ-DB SQL VSAM Work Datasets
ALOCSQLV	Allocate EZ-DB SQL VSAM Work Datasets
RUNSQLG	Process SQL Text
RUNTRC2	Consolidate SQL into Workload
RUNPATH	Generate Access Paths with Dynamic EXPLAIN
SUMMARY	Accumulate Live Workload Summary
COMPRESS	Compress CATSTATS

RUNSTATD	Dump Catalog Statistics
DELGRPTS	Delete Reports for Tracer DELRPT1
REPORTS	Generate Reports for Tracer
DELSQLV2	Delete Work Datasets

For further details of these job steps refer to the Appendix.

2.5 Refresh SQL access paths and Statistics

This option is used when you wish to add or refresh access paths and catalog statistics for SQL in an existing trace workload.

Access Paths for SQL captured in the trace may be obtained by explaining the SQL when the trace is terminated. By default, only NEW SQL where the Access Path is not already present will be processed but you may opt for ALL SQL to be explained. This will generate Version E of the Access Path Data. Refer to the Start Cache SQL Analysis option Access Path Controls.

If you do not choose to generate access paths when the trace terminates, or you wish to refresh the access paths and/or catalog statistics, you can do so using this option.

In addition, you may use this option to load or refresh the access paths for static SQL from the applicable BIND PLAN table. This will populate Version B of the access path.

The following panel is displayed:

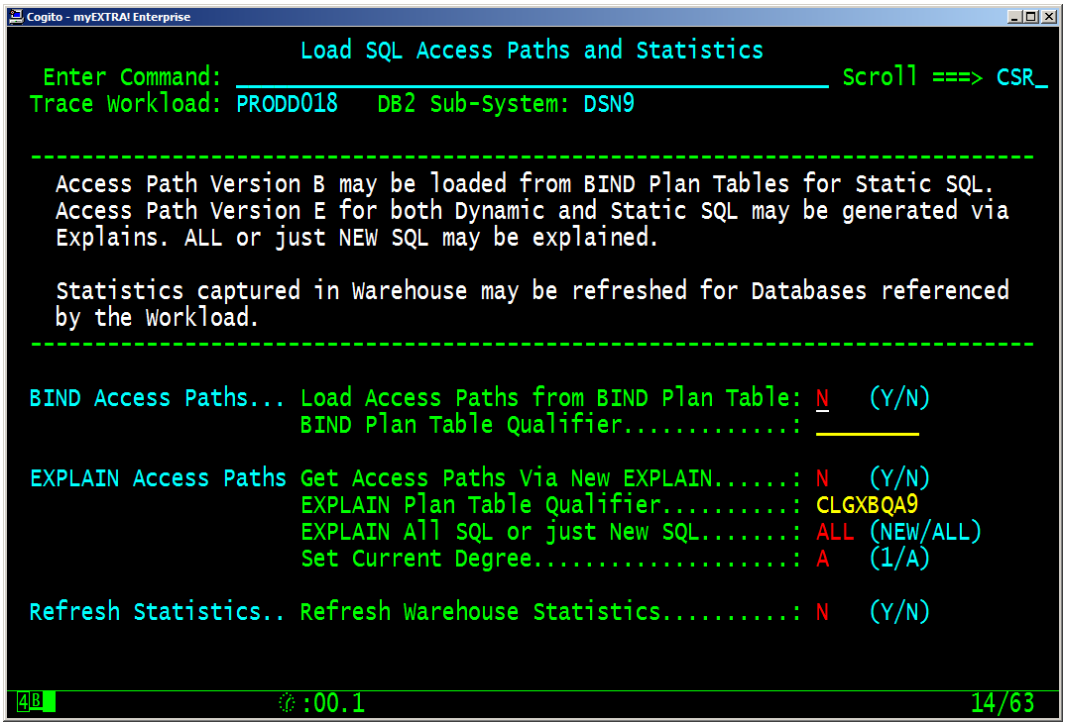


Figure 36 Generate Access Path and Cost data

The following parameters may be specified-

BIND Access Paths

Load Access Paths From BIND Plan Table

Specify whether you wish to get access paths from Plan table(s) used at BIND Time.

Y - Specify Y If you want to get the access paths from the Plan Table used at BIND time. You must have specified EXPLAIN(YES) when performing the BIND. This will load or refresh the current Version B access path for each SQL in the workload.

SQL not found in the BIND Plan Table will be Explained now (if you have also specified Get Access Paths via New Explain). This will load or refresh the current Version E access path for each SQL in the workload

N - Specify N if you do not want to get access paths from a BIND Plan table.

BIND Plan Table Qualifier

Specify the applicable qualifier(s) to be used to identify the BIND Plan table(s) to be used.

DRILL <PF4> be presented with a list of available PLAN_TABLEs from which you can select up to 15.

EXPLAIN Access Paths

Get Access Paths via New Explain

Specify whether you wish to get access paths by explaining the SQL.

Y - Specify Y to Explain the SQL using the specified PLAN table. This will load or refresh the current Version E access path for each SQL in the workload.

N - Specify N if you don't want to explain the SQL.

This is the default value.

EXPLAIN Plan Table Qualifier

Specify the applicable qualifier to be used to identify the PLAN_TABLE to be used for the explain.

DRILL <PF4> to be presented with a list of available PLAN_TABLEs from which you can select the applicable qualifier

EXPLAIN All SQL or just New SQL

NEW - Only new SQL where the access path is not already present will be explained. This is the default value.

ALL - Explain all SQL.

Set Current Degree

A — Set CURRENT DEGREE to ANY, prior to issuing the explains. This will allow parallelism, if parallelism is enabled for the subsystem.

1 — Leave CURRENT DEGREE as default 1, to disable parallelism.

Refresh Statistics

Refresh Warehouse Statistics

(Y/N)

At the end of each trace, statistics are captured to the statistics repository (warehouse) for each new database encountered. These are the statistics that are displayed when a user drills to a database object (e.g. table or index) from the access path display.

The access paths may be generated at the end of the trace, or explicitly using option EZ-Cache option 1.3 or EZ-Tracer option 2.5 to refresh the access paths.

By default, the statistics are captured for a given workload only the first time that the database is encountered. This means that it is possible for the statistics to become *stale*, and possibly not correspond to the statistics that were present at the time the access path was generated.

The Refresh Catalog statistics option in EZ-Cache and EZ-Tracer may be used to refresh the statistics each time the access paths are generated.

Y — Specify Y to force all statistics to be refreshed for the workload at the end of the trace.

N — The statistic will not be refreshed. They will only be extracted the first time the database is encountered for this workload/

Note the option to EXPLAIN all SQL or just NEW SQL. To ensure that all access paths for all SQL are in synch with the statistics, then you should EXPLAIN ALL SQL and refresh the statistics.

Press **<enter>** to display the generated job stream to generate the Access Path data.

- Type SUBMIT in the command line to submit the job for execution.
- Press the **<PF3>** key to return to the Tracer menu.

If you have received Explain failures the job will terminate with RC=4 and you should examine the log file for further information about the statements that have not been explained successfully.

Job Step Summary

The generated job contains the following internal job steps:

Job Step	Description
ALLOCDS	Allocate required datasets
RUNPATH	Generate Access Path data for all SQL in Workload

For further details of these job steps refer to the Appendix.

2.6 Access Path Change Management

It is possible to perform Impact Analysis directly on EZ-Tracer Workloads using the EZ-Impact Analyzer product. Select Option 2.6 to display the Impact Analyzer for Tracer Workloads menu as shown in the following figure:

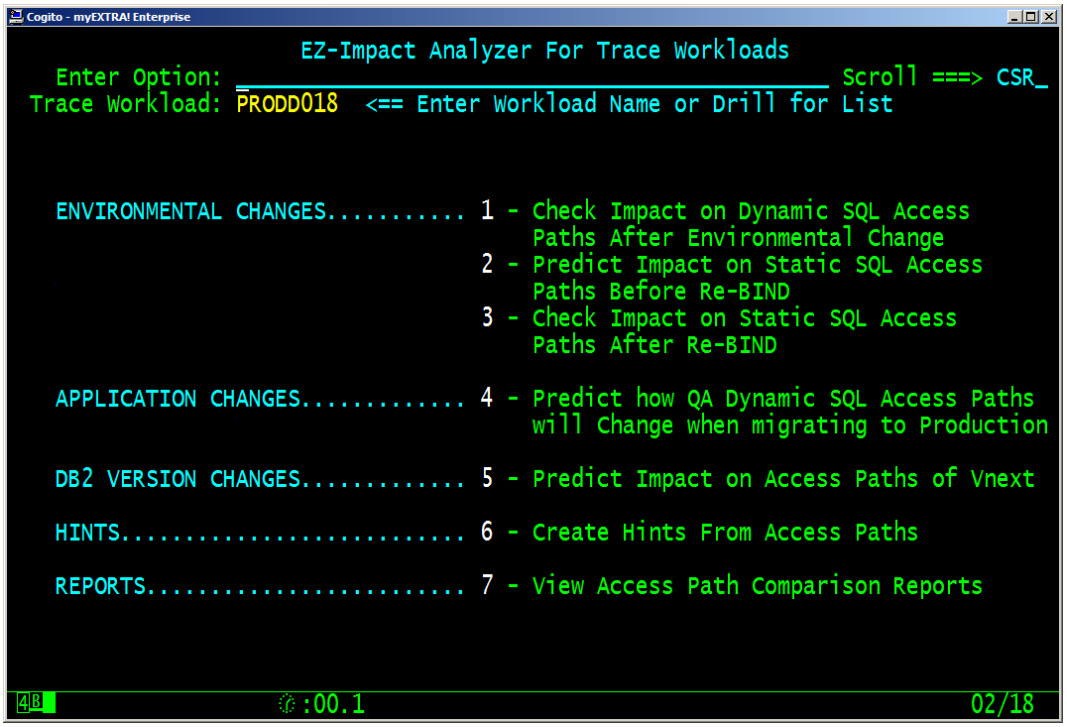


Figure 37 Impact Analyzer for Trace Workloads

The functions on the Analyze Traced SQL menu are divided into the following groupings:

Environmental Changes

1. For Dynamic SQL, Check the impact of an environmental change by comparing a new explain with the existing (Version E) access path.
2. For Static SQL, predict the impact of an environmental change by comparing a new explain with the existing (Version B) access path.
3. For Static SQL, check the impact of an environmental change after a re-bind by comparing the new bind access path with the existing (Version B) access path.

Application Changes

4. For Dynamic SQL, predict the impact of migrating to Production. Compare new Explain in the target subsystem with the existing (Version E) access path.

DB2 Version Changes	5. For Static and/or Dynamic SQL, predict the impact of migrating to DB2 Vnext. Perform new explain in the Vnext environment and compare with existing Version E or Version B access paths for Dynamic and/or Static SQL respectively.
Hints	6. Make Hints from Access Paths.
Reports	7. Display the Reports menu.

Refer to the EZ-Impact Analyzer User Guide for further information. Note you must be licensed for the EZ-Impact Analyzer product.

2.7 Set Up Scheduled Performance Trace

It is possible to set up an EZ-Tracer trace to be scheduled for automatic submission by a job scheduling system. For example, you may wish to submit a 5-minute trace every hour on the hour.

Select option 2.7 to display the Start Performance SQL trace panel as shown in the following figure:

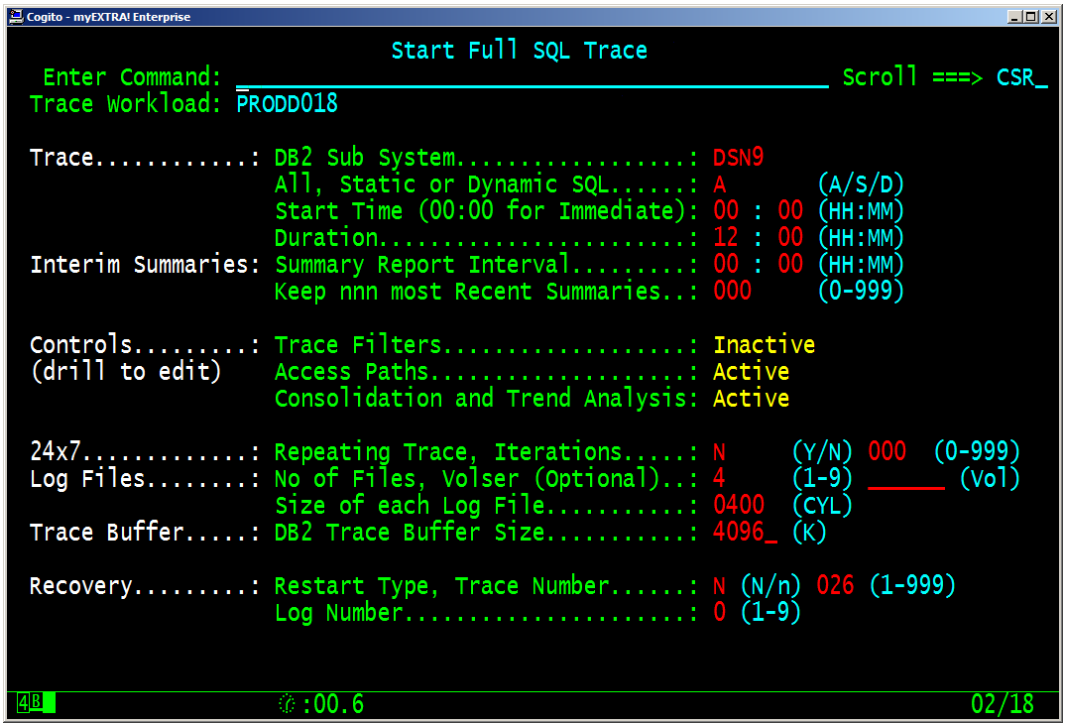


Figure 38 Start Performance Trace

Specify the required trace options exactly as if you were submitting the trace manually. See option 2.1 - Start SQL Performance Trace for further details.

For example, if you want to run a five-minute trace every hour, set Trace Interval to 00:05.

After you have selected the required parameters press <**enter**> to generate the job stream.

You must edit the generated job stream to change *your.jobsched.hlq* to an appropriate JCLLIB where the generated job will be saved.

This job stream contains the following steps.

ALLOCDS	Allocate the required datasets for the workload
RUNVSAM3	Allocate and initialise VSAM report datasets
CPYJOB18	Copy Tracer Job to specified JCLLIB

- TYPE SUB in the command line to submit the job
- <**PF3**> to return to the EZ-Cache main menu.

*The generated job will create the required EZ-Cache job stream in a member in **JCLLIB** with a member name matching the current workload name.*

See also option 2.8 for further details of submitting the job using your automated job scheduling system.

2.8 Scheduler Job

Select this option (option 2.8) to generate the job stream that is to be submitted by your job scheduling system.

You must edit the generated job stream to change *your.jobsched.hlq* to the name of the JCLLIB where the Job stream was created for this workload using option 2.7.

Do not submit this job manually unless you want a trace to start immediately. This job is to be run by your scheduler. It will initiate the next instance of the EZDB2 regular Trace job every time it is run.

This job stream contains the following steps:

EDITJOB	EDIT the member in JCLLIB to create the next trace invocation for this workload.
CPYJOB18	Copy job to the internal reader to start the next trace.

Ensure the generated JCL points to the correct JCLLIB and save this job where it can be submitted from your job scheduler.

The member in JCLLIB (created using option 2.7) contains the job stream that will be tailored and submitted to the internal reader to start the next trace. You may wish to move this from JCLLIB to some other library where it will not be overwritten.

2.9 Set Up Started Task Trace

It is possible to set up an EZ-Tracer trace to be executed as a STARTED TASK.

Select option 2.9 to display the Start Performance SQL trace panel as shown in the following figure:

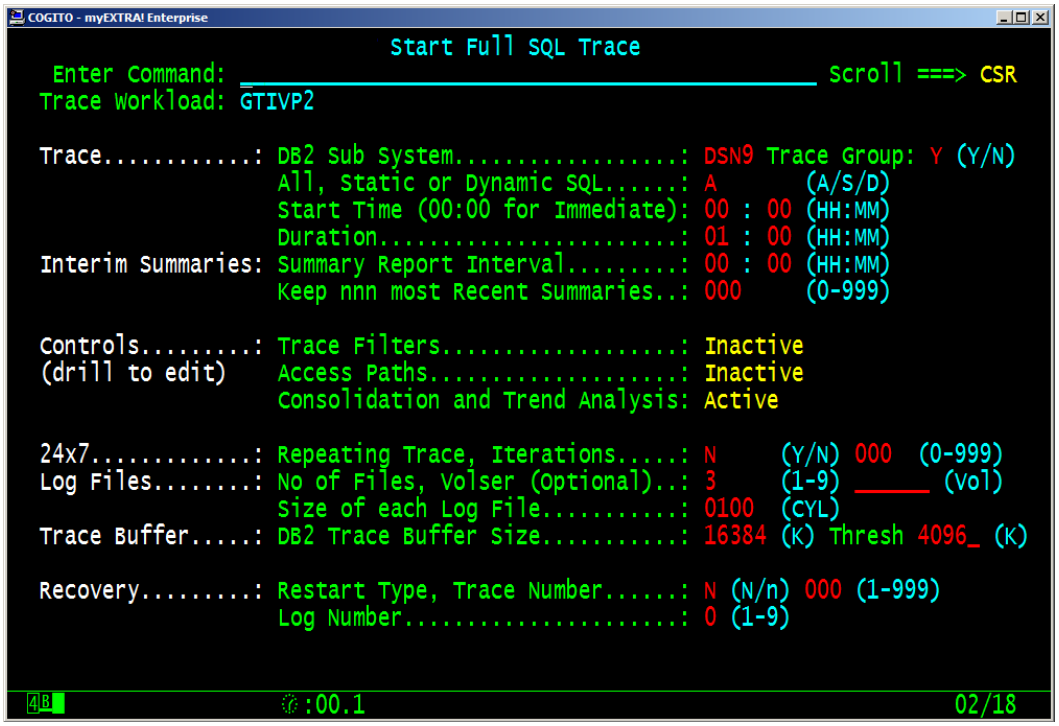


Figure 39 Set up Started Task Trace

Specify the required trace options exactly as if you were submitting the trace manually. See option 2.1 - Start SQL Performance Trace for further details.

After you have selected the required parameters press <**enter**> to generate the job stream.

The generated job stream must be edited as follows:

SET PROCLIB=your.proclib	Change *your.proclib* to the name of the JCL library that will contain the generated job stream, e.g. SYS1.PROCLIB
SET PROC=*wlname*	Change *wlname* to the name you want to use as the PROC name. The default name is the workload name.

This job stream contains the following steps.

ALLOCDS	Allocate the required datasets for the workload
RUNVSAM3	Allocate and initialise VSAM report datasets.
CPYJOB15	Convert Trace job to PROC and add it to the specified PROCLIB

- TYPE SUB in the command line to submit the job
- <**PF3**> to return to the EZ-Cache main menu.

The saved job stream may now be executed as started task in the normal way, for example using the SDSF /**S** command or using a job scheduler.
Each time the job is started it will automatically increment the trace number to initiate the next iteration of the trace.

Appendix

A Appendix

A.1 Start SQL Trace – Job Steps

XOPTRC1- Capture SQL Trace

```
+DB200110 EZ-DB2 STARTED CAF
+DB200115 EZ-DB2 Attaching Command Processor
+DB200110 EZ-DB2 STARTED TRACE DEST(OP1 )
+DB200130 EZ-DB2 Command Processor Starting
+DB200130 Start of Trace 001
+DB200110 EZ-DB2 STARTED READA
+DB200111 EZ-DB2 TRACE READ 000001406K IFCIDS AFTER 00010 MINUTES
+DB200117 EZ-DB2 KCV In Use 00006 Max 00008
+DB200118 EZ-DB2 CUR In Use 00002 Max 00008
+DB200120 End of Interval   1 Switching to LOG 02
+DB200123 Process Log 01 Initiated
+DB200111 EZ-DB2 TRACE READ 000002626K IFCIDS AFTER 00020 MINUTES
+DB200117 EZ-DB2 KCV In Use 00010 Max 00014
+DB200118 EZ-DB2 CUR In Use 00005 Max 00015
+DB200120 End of Interval   2 Switching to LOG 03
+DB200123 Process Log 02 Initiated
+DB200124 Log File full - forced to swap
+DB200123 Process Log 03 Initiated
+DB200116 EZ-DB2 Detaching Command Processor
+DB200131 EZ-DB2 Command Processor Stopping
+DB200111 EZ-DB2 TRACE READ 000004807K IFCIDS AFTER 00030 MINUTES
+DB200117 EZ-DB2 KCV In Use 00000 Max 00014
+DB200118 EZ-DB2 CUR In Use 00012 Max 00020
+DB200120 End of Interval   3
+DB200110 EZ-DB2 STOPPED TRACE
+DB200110 EZ-DB2 STOPPED CAF
+DB200123 Process Log 04 Initiated
```

Figure 40 XOPTRC1 - JESMSGLG messages

Figure 40 shows a section of a LOGFILE produced by XOPTRC1.

This example shows a Tracer run with a Summary Interval of 10 minutes specified. After each 10-minute interval, you can see that Tracer displays a message such as the following:

```
+DB200120 End of Interval 1 Switching to LOG 02
+DB200123 Process Log 01 Initiated
```

This indicates that Tracer has switched to write to LOG 02 and that XOPTRC2 has been submitted to process the data previously written to LOG 01. You will see a succession of similar messages at the end of each Log switch interval as XOPTRC1 “cycles” through the available LOG files.

You may also see messages such as the following:

```
+DB200124 Log File full - forced to swap
+DB200123 Process Log 03 Initiated
```

In this event, Tracer has had to switch log files because it has run out of space in the current log file.

If this occurs frequently, you should review the size of the log files specified in the Start SQL Trace parameters.

You should also ensure that No of Log Files is set high enough, so that XOPTRC2 has sufficient time to process the data in a given log file, before XOPTRC1 needs to switch back to that Log.

Refer to Start SQL Trace Options for further information about Summary Intervals, No of Log Files etc.

The JESMSGLG for the trace job will contain a message such as:

```
+DB200110 EZ-XOP STARTED TRACE DEST(OP2 )
```

Where Op*n* is the trace destination name allocated.

If the trace cannot start because all available trace destinations are in use, the following message is issued:

```
+DB200113 EZ-XOP All Trace Destinations In Use
```

A maximum of 8 trace destinations may be in use at any given time. Therefore, if you receive this message, you will need to terminate another active trace destination before the EZ-DB2 trace program may be started.

The following figure shows the Output information written to the XOPLOG file.

The report echoes the Run Time Parameters specified (not shown), and reports statistics for the number of records collected for the specified IFCIDS.

```
XOPTRC1  09.14.04 :
XOPTRC1 .09.14.04 : EZ-DB2 RELEASE 9.1 APAR LEVEL: CX910163
XOPTRC1  09.14.04 :    START DBA20004    2 MINUTE TRACE ON 12-07-23 AT 09.14.04      0NNNNYYN0
XOPTRC1  09.14.04 :  g  -STA TRA(P) D(OPX) C(32)IFCID(58,74,163,3,64,59,60,61,65,66,125,221,222,
XOPTRC1  09.14.04 : 231,233) BUFSIZE(16384) TDATA(COR,CPU) PKGPROG(XOPIVP*)
XOPTRC1  09.14.04 : ***   Starting SQL Trace               ***
XOPTRC1  09.15.04 : ***   End of SQL Trace                 ***
XOPTRC1  09.15.04 :       END   DBA20004    2 MINUTE TRACE ON 12-07-23 AT 09.15.04   0NNNNYYN0
XOPTRC1  09.15.04 :
XOPTRC1  09.15.04 : ***   SQL Trace IFCID Counts           ***
XOPTRC1  09.15.04 :
XOPTRC1  09.15.04 : READA/READS      :             1
XOPTRC1  09.15.04 : Records Lost     :             0
XOPTRC1  09.15.04 : IFCID 58         :          3970
XOPTRC1  09.15.04 : IFCID 59         :          3680
XOPTRC1  09.15.04 : IFCID 60         :             0
XOPTRC1  09.15.04 : IFCID 61         :            71
XOPTRC1  09.15.04 : IFCID 63/350     :            19
XOPTRC1  09.15.04 : IFCID 64         :            19
XOPTRC1  09.15.04 : IFCID 65         :           100
XOPTRC1  09.15.04 : IFCID 66         :           100
XOPTRC1  09.15.04 : IFCID 74         :             4
XOPTRC1  09.15.04 : IFCID 125        :             0
XOPTRC1  09.15.04 : IFCID 163        :             0
XOPTRC1  09.15.04 : IFCID 221        :             0
XOPTRC1  09.15.04 : IFCID 222        :             0
```

```
XOPTRC1  09.15.04 : IFCID 231        :             0
XOPTRC1  09.15.04 : IFCID 233        :             0
XOPTRC1  09.15.04 : IFCID 247        :             0
XOPTRC1  09.15.04 : IFCID 316        :             0
XOPTRC1  09.15.04 : IFCID 317        :             0
XOPTRC1  09.15.04 : IFCID 317 Flushed:             0
XOPTRC1  09.15.04 : IFCID 401        :             0
XOPTRC1  09.15.04 : Thresh Filtered  :             0
XOPTRC1  09.15.04 :
XOPTRC1  09.15.04 : ***   SQL IFCID 58 Accumulated Totals  ***
XOPTRC1  09.15.04 :
XOPTRC1  09.15.04 : ***   SQL IFCID 58 Type INDX Totals    ***
XOPTRC1  09.15.04 : ROWS PROCESSED      :  000000000000000610
XOPTRC1  09.15.04 : ROWS LOOKED AT      :  000000000000000301
XOPTRC1  09.15.04 : ROWS DM QUALIFIED   :  000000000000004718
XOPTRC1  09.15.04 : ROWS RDS QUALIFIED  :  000000000000003680
XOPTRC1  09.15.04 : ROWS INS/UPD/DEL    :  000000000000000281
XOPTRC1  09.15.04 : PAGES SCANNED       :  000000000000001087
XOPTRC1  09.15.04 : ***   SQL IFCID 58 Type SEQD Totals    ***
XOPTRC1  09.15.04 : ROWS PROCESSED      :  000000000023235903
XOPTRC1  09.15.04 : ROWS LOOKED AT      :  000000000023235903
XOPTRC1  09.15.04 : ROWS DM QUALIFIED   :  000000000000142855
XOPTRC1  09.15.04 : ROWS RDS QUALIFIED  :  000000000000127935
XOPTRC1  09.15.04 : ROWS INS/UPD/DEL    :  000000000000000359
XOPTRC1  09.15.04 : PAGES SCANNED       :  000000000000153196
XOPTRC1  09.15.04 : ***   SQL IFCID 58 Type SEQW Totals    ***
XOPTRC1  09.15.04 : ROWS PROCESSED      :  000000000000123900
XOPTRC1  09.15.04 : ROWS LOOKED AT      :  000000000000123900
XOPTRC1  09.15.04 : ROWS DM QUALIFIED   :  000000000000118880
XOPTRC1  09.15.04 : ROWS RDS QUALIFIED  :  000000000000000240
XOPTRC1  09.15.04 : ROWS INS/UPD/DEL    :  000000000000119780
XOPTRC1  09.15.04 : PAGES SCANNED       :  000000000000001480
XOPTRC1  09.15.04 :
XOPTRC1  09.15.04 : 12-07-23 Ending            RETURN CODE -    0
```

Figure 41 XOPTRC1 Log File

XOPTRC2 - Analyze SQL Trace

This step is automatically submitted to the internal reader to execute at the end of XOPTRC1, or at the end of each Summary interval.

The XOPTRC2 process reads the trace data sequential files (LG01,LG02...LG*nn*) created by XOPTRC1 and accesses the DB2 Catalog to resolve SQL statements with the SQL text. The resolved SQL data is later written to LOGO (See XOPLOGO step).

A Resolved Statement Number is the Statement number within the Package/DBRM with which the SQL Text being executed is associated. For instance, an Open Statement (IFCID 65) does not have any SQL Text associated with it in the SQL Trace. However, there will be a Declare and possibly a Prepare in the Package/DBRM, which contains the SQL text which gets executed when the Open occurs.

Static SQL is obtained from the Package/DBRM for Executes and Open. This SQL Text can be found in SYSPACKSTMT/SYSSTMT depending on whether Packages or Plans/DBRMS are in use on the system.

Notes:

- If no DBNAMEs are explicitly selected in the Trace Filter Parameters, then ALL DBNAMEs are selected.
- The input trace file is “filtered” based upon the DBNAMEs which are selected for processing. The SQL events analyzed from the SQLLOG file, are checked to see if they access any tables that belong to the active DBNAMEs selected. If not, then the SQL event is 'lost'.
- Note that if an SQL event processes against multiple databases, then if any selected database is found, the entire SQL event IS written forward and NOT discarded.
- At the end of processing, all selected DBNAMEs found, have their distinct qualifiers written out to the Qualifier Translation Control File (XOPQUALS) used for subsequent qualifier translation where required.
- All DBNAMEs encountered, regardless of whether they were originally selected, are written out to the Select Database Name control file (XOPDBNAM) at the finish, but only those initially selected are marked as “active”. All other DBNAMEs are set “inactive”. If no DBNAMES were explicitly selected, All DBNAMEs are written out but marked as “inactive”.

The following figure (Figure 42Figure 40) shows a part of a sample report written to XOPTRC2 logfile.

The report includes:

- The total number of SQL events processed
- The total number of DBNAMES referenced by the processed SQL events and the corresponding names.
- The total number of TABLE NAMES referenced by the processed SQL events, and the corresponding names.
- If XOPTRC2 had trouble resolving any of the SQL event data included in the LG01,LG02...LGnn files, diagnostic information is included in this report.

  Note: The successfully resolved SQL event data is written to the LOGO output dataset by the subsequent XOPLOGO step.

XOPTRC2 Return Codes

RC	Description
0	Processing completed successfully.
12	Invalid XOPPARMS parameter. See LOGFILE for details.
16	SQL error occurred. See LOGFILE for SQL diagnostics.
20	IO Error on file - check job log for details
99	Unexpected error.

The following figure shows the information messages written to the XOPTRC2 log file.

```
XOPPARM  09.15.05 : EZ-DB2 Release 9.1 APAR Level: CX910163
XOPTRC2  09.15.05 : R9.10 Starting Call Attach to DB2 Subsystem DBA2
XOPTRC2  09.15.05 : DB2 Version 10.1.0 NFM
XOPTRC2  09.15.05 : (9.1152)  Starting SQL Consolidation
XOPTRC2  09.15.05 :
XOPTRC2  09.15.05 : Processing a single trace log file
XOPTRC2  09.15.05 : Consolidate SQL
XOPTRC2  09.15.05 : Consolidate AUTHIDs for non-distributed SQL
XOPTRC2  09.15.05 : Consolidate, first adding table qualifiers
XOPTRC2  09.15.05 : Consolidate differing length IN lists
XOPTRC2  09.15.05 : Consolidate DDF Program Names
XOPTRC2  09.15.05 :
XOPTRC2  09.15.06 :
XOPTRC2  09.15.06 : Processed         0 minutes of trace data,       3874 trace events.
EZX13005 09.15.06 :
EZX13005 09.15.06 : Trace Summary
EZX13005 09.15.06 : -------------
EZX13005 09.15.06 : Total Trace Events read         =       3874
EZX13005 09.15.06 : Trace Events for SQL            =        110
EZX13005 09.15.06 :
EZX13005 09.15.06 : Other Trace Events:
EZX13005 09.15.06 : SQL prepares                    =         19
EZX13005 09.15.06 : End of thread                   =          4
EZX13005 09.15.06 :
EZX13005 09.15.06 : SQL processed this workload     =         19
EZX13005 09.15.06 : SQL processed this trace        =         19
EZX13005 09.15.06 : New SQL added to workload       =          0
EZX13005 09.15.06 :
EZX13006 09.15.06 : Tables Referenced By Selected SQL = 000000003
EZX13006 09.15.06 : TableName                      DBName
EZX13006 09.15.06 : SGBLIVA2 . XOP_CUSTOMER        DGBLIVA2
EZX13006 09.15.06 : SGBLIVA2 . XOP_ORDER           DGBLIVA2
EZX13006 09.15.06 : SGBLIVA2 . XOP_PRODUCT         DGBLIVA2
EZX13006 09.15.06 :
EZX13007 09.15.06 : DBNames Referenced By Selected SQL = 000000001
EZX13007 09.15.06 : DGBLIVA2
EZX13007 09.15.06 :
XOPTRC2  09.15.06 : End of User Reports
XOPTRC2  09.15.06 : - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
XOPTRC2  09.15.06 : Internals Reports XOPTRC2  09.15.06 : Thread records  = 00000000
XOPTRC2  09.15.06 : High water mark = 00000001
XOPTRC2  09.15.06 : Allocation      = 00050000
XOPTRC2  09.15.06 : Reads           = 00000000
XOPTRC2  09.15.06 : F-reads         = 00000004
XOPTRC2  09.15.06 : Writes          = 00000004
XOPTRC2  09.15.06 : Deletes         = 00000004
XOPTRC2  09.15.06 : SQL slots       = 00000010
XOPTRC2  09.15.06 : Reallocate x2   = 00000000
XOPTRC2  09.15.06 : Reallocate x3   = 00000000
XOPTRC2  09.15.06 : Reallocate >3   = 00000000
XOPTRC2  09.15.06 : Reallocate HWM  = 00000000
XOPTRC2  09.15.06 :
XOPTRC2  09.15.06 : SQL prep. recs  = 00000019
XOPTRC2  09.15.06 : SQL exec. recs  = 00000019
XOPTRC2  09.15.06 : In use at end   = 00000038
XOPTRC2  09.15.06 : High water mark = 00000038
XOPTRC2  09.15.06 : Allocation      = 00150000
XOPTRC2  09.15.06 : Reads           = 00000173
XOPTRC2  09.15.06 : F-reads         = 00000100
XOPTRC2  09.15.06 : Writes          = 00000038
XOPTRC2  09.15.06 : Deletes         = 00000000
XOPTRC2  09.15.06 :
XOPTRC2  09.15.06 : SQL stats recs  = 00000019
XOPTRC2  09.15.06 : High water mark = 00000019
```

```
XOPTRC2  09.15.06 : Allocation      = 00150000
XOPTRC2  09.15.06 : Reads           = 00000019
XOPTRC2  09.15.06 : F-reads         = 00000019
XOPTRC2  09.15.06 : Writes          = 00000019
XOPTRC2  09.15.06 : Deletes         = 00000000
XOPTRC2  09.15.06 :
XOPTRC2  09.15.06 : Plan/Pack recs  = 00000004
XOPTRC2  09.15.06 : High water mark = 00000004
XOPTRC2  09.15.06 : Allocation      = 00050000
XOPTRC2  09.15.06 : Reads           = 00000004
XOPTRC2  09.15.06 : F-reads         = 00000004
XOPTRC2  09.15.06 : Writes          = 00000004
XOPTRC2  09.15.06 : Deletes         = 00000000
XOPTRC2  09.15.06 :
XOPTRC2  09.15.06 : PPH key recs    = 00000034
XOPTRC2  09.15.06 : High water mark = 00000034
XOPTRC2  09.15.06 : Allocation      = 00500000
XOPTRC2  09.15.06 : Reads           = 00000089
XOPTRC2  09.15.06 : F-reads         = 00000034
XOPTRC2  09.15.06 : Writes          = 00000034
XOPTRC2  09.15.06 : Deletes         = 00000000
XOPTRC2  09.15.06 :
XOPTRC2  09.15.06 : Synonym recs    = 00000003
XOPTRC2  09.15.06 : High water mark = 00000003
XOPTRC2  09.15.06 : Allocation      = 00020000
XOPTRC2  09.15.06 : Reads           = 00000030
XOPTRC2  09.15.06 : F-reads         = 00000003
XOPTRC2  09.15.06 : Writes          = 00000003
XOPTRC2  09.15.06 : Deletes         = 00000000
XOPTRC2  09.15.06 :
XOPTRC2  09.15.06 : Temp. Tab recs  = 00000000
XOPTRC2  09.15.06 : High water mark = 00000000
XOPTRC2  09.15.07 : Allocation      = 00050000
XOPTRC2  09.15.07 : Reads           = 00000000
XOPTRC2  09.15.07 : F-reads         = 00000000
XOPTRC2  09.15.07 : Writes          = 00000000
XOPTRC2  09.15.07 : Deletes         = 00000000
XOPTRC2  09.15.07 :
XOPTRC2  09.15.07 : Tokens HWM      = 00000082
XOPTRC2  09.15.07 :
XOPTRC2  09.15.07 : Pre-normalization hits  =                 2
XOPTRC2  09.15.07 : Post-normalization hits =                 0
XOPTRC2  09.15.07 : Normalization misses    =                17
XOPTRC2  09.15.07 :
XOPTRC2  09.15.07 : *** : XOPSQLCP File Records Written:                0
XOPTRC2  09.15.07 : *** : XOPSQLFP File Records Written:               20
XOPTRC2  09.15.07 : *** : XOPSQLFP File Records Rewritten:             39
XOPTRC2  09.15.07 : *** : XOPSQLTP File Records Written:                0
XOPTRC2  09.15.07 : *** : XOPSQLTP File Records Rewritten:             17
XOPTRC2  09.15.07 : *** : XOPSQLXP File Records Written:                0
XOPTRC2  09.15.07 : *** : XOPSQLVP File Records Written:                0
XOPTRC2  09.15.07 : *** : XOPSQLSP File Records Written:                0
XOPTRC2  09.15.07 : *** : XOPSQLDP File Records Written:                0
XOPTRC2  09.15.07 :
XOPTRC2  09.15.07 : *** : XOPSQLCP File Records Read   :                0
XOPTRC2  09.15.07 : *** : XOPSQLFP File Records Read   :               76
XOPTRC2  09.15.07 : *** : XOPSQLTP File Records Read   :               37
XOPTRC2  09.15.07 : *** : XOPSQLXP File Records Read   :               19
XOPTRC2  09.15.07 :
XOPTRC2  09.15.07 : R9.10 Stopping Call Attach
XOPTRC2  09.15.07 :  End of SQL Consolidation
XOPTRC2  09.15.07 : 12/07/23 Ending            RETURN CODE - 0000
XOPTRC2  09.15.07 :
```

Figure 42 XOPTRC2 Log File

- Details of the DBNAMEs read in and set active are shown If using Trace Filtering. Otherwise, you will see the message - All DBNames will be Selected.
- **SQL trace Events Processed** - Reports the total number of trace events processed.
- **Tables referenced by Selected SQL** - Reports all table names referenced by the selected SQL statements.
- **DBNames Referenced By Selected SQ**L - reports all DBNAMEs referenced at base table level. For example, on views spanning multiple databases, all the DBNAMEs are reported.

The catalog statistics for all referenced databases will be extracted to the EZ-DB2 statistics warehouse by the subsequent XOPSTATD step.

RUNPATH - Generate Access Path Data

The RUNPATH job step executes XOPPATH which generates the access path data for the traced SQL.

```
XOPPARM  09.15.07 : EZ-DB2 Release 9.1 APAR Level: CX910163
XOPPATH  09.15.07 : R9.10 Starting Call Attach to DB2 Subsystem DBA2
XOPPATH  09.15.07 : DB2 Version 10.1.0 NFM
XOPPATH  09.15.09 : ******************************************************
XOPPATH  09.15.09 : * Starting Access Path Generation                     *
XOPPATH  09.15.09 : * EXPLAIN Plan Table             : COG02              *
XOPPATH  09.15.09 : * Access Path Version            : E                  *
XOPPATH  09.15.09 : ******************************************************
XOPPATH  09.15.09 :
XOPPATH  09.15.12 :
XOPPATH  09.15.12 : R9.10 Stopping Call Attach
XOPPATH  09.15.12 :
XOPPATH  09.15.12 :
XOPPATH  09.15.12 :  Total SQL Alerts Raised                              0
XOPPATH  09.15.12 :
XOPPATH  09.15.12 :  Total Static SQL              :      0
XOPPATH  09.15.12 :  Total SQL in BIND Plan Table  :      0
XOPPATH  09.15.12 :  Total Static SQL Explained    :      0
XOPPATH  09.15.12 :  Total Dynamic SQL             :     19
XOPPATH  09.15.12 :  Total Dynamic SQL Explained   :      0
XOPPATH  09.15.12 :  Total SQL Already Explained   :     19
XOPPATH  09.15.12 :  Total Inserts Processed       :      0
XOPPATH  09.15.12 :  Total Inserts No Unique Ix    :      0
XOPPATH  09.15.12 :  Total Explain Failures        :      0
XOPPATH  09.15.12 :  Total SQL Can't Be Explained  :      0
XOPPATH  09.15.12 :  Total Temp. Tables Declared   :      0
XOPPATH  09.15.12 :  End of   Access Path Generation
XOPPATH  09.15.12 : 12/07/23 Ending            RETURN CODE – 0000
```

Figure 43 XOPPATH Log file

The XOPLOG file will contain information such as

- the number of access paths obtained from BIND plan tables
- the number of access paths obtained by Explain
- the number of Explain failures

If Explain failures have occurred the LOG will also include information about the SQL that failed to explain, and the job will terminate with RC=4.

SUMMARY - Summarize Workload

After the final iteration of XOPTRC2 this step is invoked to execute the XOPSUMM process. This process accumulates the workload summary data for display by the Online reporting function.

RUNSTATD - Extract System Catalog Statistics

XOPSTATD is executed to extract the System Catalog statistics for all traced databases to the Statistics Warehouse. The statistics may be subsequently used in the EZ-DB2 Analysis system if the Trace Workload is to be analysed by other EZ-DB2 components such as EZ-Alerts, EZ-Index Optimizer etc. You may view the statistics in the Statistics Warehouse using the EZ-Stats Statistics manager.

The following figure shows a sample of the information in the XOPSTATD log file:

```
XOPPARM  09.15.16 : EZ-DB2 Release 9.1 APAR Level: CX910163
 XOPSTATD 09.15.16 : R9.10 Starting Call Attach to DB2 Subsystem DBA2
 XOPSTATD 09.15.16 : DB2 Version 10.1.0 NFM
 XOPSTATD 09.15.16 : R9.10 XOPSTATD - Dump Catalog Statistics Starting -------
 XOPSTATD 09.15.16 : Unloading database name DGBLIVA2
 XOPSTATD 09.15.18 : XOPSTATD  :    147 Output Records Written
 XOPSTATD 09.15.18 : XOPSTATD  :     03 Rows Read From SYSIBM.SYSTABLESPACE
 XOPSTATD 09.15.18 : XOPSTATD  :     03 Rows Read From SYSIBM.SYSTABLES
 XOPSTATD 09.15.18 : XOPSTATD  :     00 Rows Read From SYSIBM.SYSTABSTATS
 XOPSTATD 09.15.18 : XOPSTATD  :     03 Rows Read From SYSIBM.SYSINDEXES
 XOPSTATD 09.15.18 : XOPSTATD  :     12 Rows Read From SYSIBM.SYSCOLUMNS
 XOPSTATD 09.15.18 : XOPSTATD  :    104 Rows Read From SYSIBM.SYSCOLDIST
 XOPSTATD 09.15.18 : XOPSTATD  :     03 Rows Read From SYSIBM.SYSTABLEPART
 XOPSTATD 09.15.18 : XOPSTATD  :     03 Rows Read From SYSIBM.SYSINDEXPART
 XOPSTATD 09.15.18 : XOPSTATD  :     03 Rows Read From SYSIBM.SYSINDEXSPACESTATS
 XOPSTATD 09.15.18 : XOPSTATD  :    131 Rows Read in Total
 XOPSTATD 09.15.18 : R9.10 Stopping Call Attach
 XOPSTATD 09.15.18 : R9.10 XOPSTATD - Dump Catalog Statistics Completed ------
 XOPSTATD 09.15.18 : 12/07/23 Ending            RETURN CODE – 0000
```

Figure 44 XOPSTATD LOG file

Refer to the EZ-Stats User Guide for further details about EZ-Stats and XOPSTATD processing.

DELGRPTS - Delete Workload Reports

In order to increase the speed of the online reporting process - the reports for the workload are pre-built into report files for the current workload. Each time a new trace is run, all the existing reports for this workload are deleted, as they will be re-built in the subsequent step to take into account the new trace data. Note that you can manually view and/or delete specific reports for a particular workload using the Workload Reports menu option (9.7)

REPORTS - Create Workload Reports

See previous step DELGRPTS. The report files for a workload are pre-built into report members for the workload each time a new trace is executed. The reports may then be browsed online using the online reporting function.

Note that you can manually view and/or delete specific reports for a particular workload using the Workload Reports menu option (9.7)

A.2 EZTREXT1 – User Exit

EZ-Tracer has an optional user exit – EZTREXT1, which allows the captured trace data to be modified prior to being written forward to the EZ-Tracer log file.

A sample user exit source module EZTREXT1 is delivered in the EZDB2 runtime SRCLIB. This should be tailored to the site's requirements, before being assembled and link edited into the EZDB2 runtime loadlib.

Install JCL

The exit is a standard assembler routine and can be assembled and link edited using the sample JCL below:

```
//    JOB
//*
//*********************************************************************
//*  Assemble EZTREXT1 Tracer User Exit                               *
//*********************************************************************
//ASMEXT1 EXEC PGM=ASMA90,PARM=(OBJECT,NODECK,'SIZE(MAX,ABOVE)')
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//         DD  DISP=SHR,DSN=ezdbspfx.SRCLIB
//SYSLIN   DD  DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(800,(500,500)),DCB=BLKSIZE=800
//SYSUT1   DD  SPACE=(800,(500,500)),UNIT=SYSDA,
//             DISP=(,DELETE)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  DISP=SHR,DSN=ezdbspfx.SRCLIB(EZTREXT1)
//*
//*********************************************************************
//*  Link Edit EZTREXT1                                               *
//*********************************************************************
//LNKPASS EXEC PGM=IEWL,PARM='XREF,AMODE(31),RMODE(ANY)',COND=(4,LT)
//SYSUT1   DD  SPACE=(800,(500,500)),UNIT=SYSDA,
//             DISP=(,DELETE)
//SYSLIN   DD  DSN=&&LOADSET,DISP=(OLD,PASS)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DISP=SHR,DSN=ezdbspfx.LOADLIB
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 ENTRY EZTREXT1
 NAME EZTREXT1(R)
//*
```

A return code of '4' for the assemble/link edit is acceptable.

The log record passed to the exit is mapped by the supplied #SQLLOG DSECT. Details of the parameters passed to the exit can be found in the supplied sample code. The sample exit as supplied has an example that translates the AUTHID field, prior to writing out the log record.

Once the exit has been successfully assembled and link edited into the EZDB2 runtime loadlib, it can be activated for a given workload/trace by drilling to the 'Trace Filters' panel.

Refer to Trace Filters parameter Activate User Exit.

When the trace starts, a message is output indicating that the user exit is active:

```
DB200141 User exit EZTREXT1 Active
```

Sample Exit

```
        TITLE 'EZ-TRACER SAMPLE USER EXIT'
*********************************************************************
*                                                                   *
*        ENTRY POINT : EZTREXT1                                     *
*        LINKAGE     : AMODE 31 RMODE ANY                           *
*                                                                   *
*        DESCRIPTION : EZ-TRACER USER EXIT 1                        *
*                      Called by EZ-Trace to allow manipulation of  *
*                      trace data before being written to the SQL Log*
*                      file for consolidation processing.           *
*                                                                   *
*        PARAMETERS  : ON ENTRY R1 POINTS TO PARAMETER LIST         *
*                      PARM1     DS  XL4   FLAG BYTES               *
*                      PARM2     DS  A     ADDRESS OF WORK AREA (4K) *
*                      PARM3     DS  A     ADDRESS OF SQLLOGO RECORD *
*                                                                   *
*        PARM 1 FLAGS: XL1       '01'  CALL IS FOR SQLLOGO RECORD   *
*                      XL3       SPARE                              *
*                                                                   *
*        OPERATION   : ON ENTRY R13 POINTS TO A STANDARD REGISTER   *
*                      SAVE AREA. EZTREXT1 MUST SAVE AND RESTORE    *
*                      REGISTERS ON ENTRY AND EXIT.                 *
*                      THE SQL LOG RECORD IS MAPPED BY THE SUPPLIED *
*                      DSECT #SQLLOG AND CONTAINS THE RECORD TYPE,  *
*                      RECORD DATA.                                 *
*                                                                   *
*        THIS CODE IS A SAMPLE USED TO MODIFY AUTHID.               *
*        THE AUTHID IS CHECKED AND MODIFIED ACCORDING TO THE        *
*        FOLLOWING TABLE:                                           *
*                                                                   *
*        AC%    ->AC_GRP      Assistance Center                     *
*        DC%    ->DC_GRP      Data Control                          *
*        HRPRD% ->PSOFT       PSOFT AppServers                      *
*        HR%    ->HR_GRP      Human Resources                       *
*        PS???% ->PSOFT       PSPRD* PSLMF* Servers                 *
*        TS%    ->TS_GRP      Tech Support                          *
*                                                                   *
*********************************************************************
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
```

```
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
EZTREXT1 CSECT
        USING EZTREXT1,R15
         B     STRT0000              * BRANCH ROUND CONSTANTS
         DC    CL8'EZTREXT1'
         DC    C'&SYSDATE '
         DC    C'&SYSTIME '
         DC    C'COPYRIGHT COGITO LTD 2011'
        DROP  R15
        USING EZTREXT1,R12
STRT0000 DS    0H
         STM   R14,R12,12(R13)       * Save callers registers
         LR    R12,R15               * Set base reg
         L     R11,4(,R1)            * Address working storage
         L     R2,8(,R1)             * Address SQL Log record
*
        USING SQLLOG,R2
        USING WORKDS,R11
         ST    R13,RSA+4             * SAVE BACKWARD CHAIN
         LA    R14,RSA               * SET FORWARD CHAIN
         ST    R14,8(R13)            * IN SAVE AREA
         LR    R13,R14               * SET OUR SAVE AREA
*********************************************************************
***      PROCESS SQLLOG RECORD                                      *
*********************************************************************
MAIN0000 DS    0H
         CLI   0(R1),X'01'           * IS THIS FOR SQLLOGO?
         BNE   RTRN0000              * NO USER PROCESSING
         CLC   SQLRECIF,=CL3'IFC'    * IFCID DATA RECORD?
         BNE   RTRN0000              * NO USER PROCESSING
*********************************************************************
***      SEARCH WFAUTTAB FOR PATTERN MATCH ON AUTHID                *
*********************************************************************
MAIN0020 DS    0H
         LA    R1,WFAUTENT         ADDRESS 1ST WFAUTENT ENTRY
        USING WFAUTENT,R1
         LA    R3,WFAUT#           COUNT OF ENTRIES IN THIS TABLE
         LTR   R3,R3               TABLE EMPTY?
         BZ    RTRN0000            YES, SO DON'T LOOK
*********************************************************************
*** CHECK FOR MATCH ON AUTHID. R1 POINTS TO CURRENT ENTRY IN TABLE.
*********************************************************************
MAIN0040 DS    0H
         LA    R15,L'WFAUTPAT      AUTHID LENGTH
         LA    R4,SQLIAUTT         AUTHID IN SQL LOG RECORD
         LA    R5,WFAUTPAT         CURRENT PATTERN TO TEST
MAIN0050 DS    0H
         CLI   0(R5),C'%'          MATCH ALL REMAINING ?
         BE    MTCH0000            YES, MATCH FOUND
         CLI   0(R5),C'?'          IGNORE THIS CHARACTER ?
         BE    MAIN0060            YES
         CLC   0(1,R4),0(R5)       CHARACTERS MATCH ?
         BNE   MAIN0070            NO, SKIP TO NEXT WFAUTENT
MAIN0060 DS    0H
         LA    R4,1(,R4)           BUMP CHAR POINTERS.
```

```
         LA    R5,1(,R5)           *
         BCT   R15,MAIN0050        ALL CHARS MATCHED?
         B     MTCH0000            YES, MATCH FOUND
MAIN0070 DS    0H
         LA    R1,WFAUTELN(,R1)    BUMP TO NEXT ENTRY IN TABLE
         BCT   R3,MAIN0040         CONTINUE SEARCHING
         B     RTRN0000            NOT FOUND, JUST RETURN.
*
MTCH0000 DS    0H
         MVC   SQLIAUTL,WFAUTNLN     * MOVE NEW AUTHID LENGTH TO SQLLOG
         MVC   SQLIAUTT,WFAUTNEW     * MOVE NEW AUTHID TEXT TO SQLLOG
*
RTRN0000 DS    0H
         SR    R15,R15               * Return code
         L     R13,4(,R13)           * Restore caller's R13
         L     R14,12(,R13)          * Restore R14
         LM    R0,R12,20(R13)        * Restore R0-R12
         BR    R14                   * and return
*
         LTORG
*********************************************************************
* Authid Translation Table
* Provided to allow users to perform their own consolidation on AUTHID
* to reduce then volume of data to be managed.
* Match on AUTHID pattern and replace with the new AUTHID value.
* Note AUTHID is always 8 characters padded with spaces.
*********************************************************************
WFAUTTAB DS    0F                    * AUTHID TRANSLATION TABLE
WFAUTENT DS    0H                    * AUTHID TRANSLATION ENTRY
WFAUTPAT DC    CL8'AC%     '         * AUTHID PATTERN TO MATCH
WFAUTNEW DC    CL8'AC_GRP  '         * REPLACEMENT AUTHID VALUE
WFAUTNLN DC    H'8'                  * REPLACEMENT AUTHID TEXT LENGTH
WFAUTELN EQU *-WFAUTENT              * LENGTH OF TABLE ENTRY
         DC    CL8'DC%     '
         DC    CL8'DC_GRP  '
         DC    H'8'
         DC    CL8'HRPRD%  '
         DC    CL8'PSOFT   '
         DC    H'8'
         DC    CL8'HR%     '
         DC    CL8'HR_GRP  '
         DC    H'8'
         DC    CL8'PS???%  '
         DC    CL8'PSOFT   '
         DC    H'8'
         DC    CL8'TS%     '
         DC    CL8'TS_GRP  '
         DC    H'8'
WFAUT#   EQU (*-WFAUTTAB)/WFAUTELN   * NUMBER OF TABLE ENTRIES
*
*        DSECTS
*
WORKDS   DSECT
RSA      DC    18F'0'                * OUR RSA
*
WORKLEN  EQU  *-WORKDS
*
         COPY  #SQLLOG
```

```
          END   EZTREXT1
```

#SQLLOG –DSECT

```
*********************************************************************
*   #SQLLOG: SQLLOG FILE                                            *
*            VARIABLE LENGTH DEPENDING ON RECORD TYPE AND CONTENT.  *
*      Type: START  = TRACE HEADER                                  *
*            IFC316 = DYNAMIC STATEMENT CACHE SQL STATISTICS        *
*            IFCNNN = ANY OLD IFCID STATISTICS RECORD               *
*            SQLTXT = SQL TEXT RECORD                               *
*            END    = END OF TRACE                                  *
*                                                                   *
*********************************************************************
*  Number of ROWSTATS entries changed from 12 to 6                  *
*  IFCID 316 and regular IFCIDS use same format                     *
*********************************************************************
SQLLOG   DSECT
*********************************************************************
*  Prefix for all Trace records                                     *
*********************************************************************
SQLTRPFX DS    0D                  * Trace Record Prefix
SQLRECLN DS    H                   * Record Length
         DS    H                   * Reserved
SQLCDIND DS    XL2                 * Compressed/Decompressed
SQLCOMPR EQU   X'0001'             *  Data Compressed
SQLRECID DS    CL6                 * Record Type
*SQLRSTRT EQU  C'START '           *  Trace Header
*SQLRTEXT EQU  C'SQLTXT'           *  SQL Text
*SQLRI316 EQU  C'IFC316'           *  DSC
*SQLREND  EQU  C'END   '           *  Trailer
*SQLRINNN EQU  C'IFCNNN'           *  IFCID nnn
         ORG    SQLRECID
SQLRECIF DC    C'IFC'              * If IFCID
SQLIFCID DS    CL3                 * IFCID NUMBER
SQLLDATA DS    0F                  * Start of Trace Record Data
SQLPFXLN EQU   *-SQLLOG            * Length of Prefix
*********************************************************************
*  Trace Header Record                                              *
*********************************************************************
SQLHDR   DS    0F                  * Trace Header Record
SQLHSSID DS    CL4                 * DB2 SSID
SQLHTRNO DS    CL4                 * Trace Number
         DS    C                   *
SQLHDURA DS    CL4                 * Trace Duration in minutes
         DS    C                   *
         DC    CL12'MINUTE TRACE'  *
         DS    C' ON '             *
SQLHDATE DS    CL8                 * Trace Start Date
         DS    C' AT '             *
SQLHTIME DS    CL8                 * Trace Start Time
SQLHISNO DS    XL2                 * History Interval Number
SQLHINT# DS    XL2                 * Number of Intervals
SQLHMINS DS    XL2                 * History Interval Minutes
SQLHRSTR DS    X                   * Trace Restart
SQLHCONT DS    X                   * Continuous Trace Ind
SQLH24X7 EQU   C'Y'                * Continuous Trace
SQLHPOSI DS    X                   * Trace from first Post Ind
```

```
SQLHPOST EQU   C'Y'                * Trace from first Post
SQLHEOJI DS    X                   * Stop Trace at EOJ Ind
SQLHEOJ  EQU   C'Y'                * Stop Trace at EOJ
SQLHDSCI DS    X                   * Trace DSC Ind
SQLHDSC  EQU   C'Y'                * Trace DSC
SQLHCPUI DS    X                   * Trace CPU Ind
SQLHCPU  EQU   C'Y'                * Trace CPU
SQLHSTAI DS    X                   * Trace Statistics Ind
SQLHSTAT EQU   C'Y'                * Trace Statistics
SQLHASQI DS    X                   * Generated by ASQL Ind
SQLHASQL EQU   C'Y'                * Generated by ASQL
SQLHDEFF DS    X                   * Default SQL Frequency
         DS    CL30                * Some Filler.
SQLHLEN  EQU   *-SQLLOG            * Length of Trace header data
*********************************************************************
*  IFCID Data Record                                                *
*********************************************************************
         ORG   SQLLDATA
SQLIFKEY DS    0C                * IFCID Key
SQLITHRD DS    CL22              * Thread Key
SQLICTKN DS    CL8               * Contoken
         DS    CL2               * Filler
         ORG   SQLIFKEY
SQL316ID DS    CL20              * DSC Statement ID
SQL316TM DS    CL10              * DSC Date/Time Inserted in Cache
         DS    XL2               * Filler
*
SQLIPROG DS    CL18              * Program Name
SQLIPLAN DS    0CL10             * Plan Name
SQLIPLNL DS    XL2               * Plan Name Len
SQLIPLNT DS    CL8               * Plan Name Text
SQLIAUTH DS    0CL10             * AUTHID
SQLIAUTL DS    XL2               * AUTHID Len
SQLIAUTT DS    CL8               * AUTHID Text
SQLICORR DS    CL12              * Correlation ID
SQLIWKST DS    CL18              * Workstation
SQLICOLL DS    CL18              * Collection
SQLICURS DS    CL18              * Cursor Name
         ORG   SQLICOLL
SQLISETC DS    CL8               * IFCID 316 Current SQLID
SQLIQUAL DS    CL8               * IFCID 316 Default Qualifier
SQLIDSCF DS    XL2               * IFCID 316 Flag
         DS    XL18              * IFCID 316 Filler
         ORG
SQLISTYP DS    CL8               * Statement Type
SQLIESTN DS    XL4               * Execute Statement No
SQLIPSTN DS    XL4               * Prepare Statement No
*
SQLSTATS DS    0F
SQLISSTK DS    XL8               * SQL Start Time
SQLICPU  DS    XL8               * SQL CPU
SQLISTK  DS    XL8               * SQL Clock
SQLIFFRQ DS    XL8               * SQL Fetch Frequency
SQLIFCPU DS    XL8               * SQL Fetch CPU
SQLIFSTK DS    XL8               * SQL Fetch Clock
SQLIFREQ DS    XL4               * SQL Trace Frequency
SQLITXT# DS    XL2               * SQL Text Record Count
SQL163TY DS    C                 * IFCID 163 Type
```

```
SQLIPOPT DS    X                  * Parser Options
SQLIREOP DS    XL2                * Reopt Vars
SQLUNICD DS    XL1                * IFCID Unicode Flag
SQLEXTND DS    XL1                * SQL Extension Data follows
SQLEX906 EQU   C'Y'                 Y = IFCID 906 Only
SQLEX247 EQU   C'H'                 H = IFCID 247 Only
SQLEXBTH EQU   C'B'                 B = IFCID 247 and 906
SQLSTATL EQU   *-SQLSTATS         * Length of Stats area
SQLIFCLN EQU   *-SQLLOG           * Length of Fixed IFCID Data
*********************************************************************
*  IFCID Data Record VARCHAR Sections                               *
*********************************************************************
*
*********************************************************************
*  IFCID Row Statistics                                             *
*********************************************************************
SQLROWS  DS    0F                 * Row Statistics
SQLROWLN DS    XL2                * Length of Rowstats Data
SQLROWE# DS    XL2                * No of Rowstats Entries
SQLRSHLN EQU   *-SQLROWE#         * Length of Rowstats Header
*
SQLROWE  DS    0F                 * Rowstats Entry
SQLRID   DS    C                  * Rowstats ID I=IFCID F=FETCH
SQLRTYPE DS    C                  * Rowstats Type X,D,W,C
         DS    CL2                * Filler
SQLRTOTS DS    6XL4               * Rowstats
         ORG    SQLRTOTS
SQLROWPR DS    F                  * Rows Processed
SQLROWLA DS    F                  * Rows Looked At
SQLROWS1 DS    F                  * Rows Stage 1
SQLROWS2 DS    F                  * Rows Stage 2
SQLROWUP DS    F                  * Rows Updated
SQLROWGP DS    F                  * Getpages
SQLRSELN EQU   *-SQLROWE          * Length of Rowstats Entry
         DS    5CL(SQLRSELN)      * 5 more Rowstats Entries
*********************************************************************
*  IFCID Long Column Names - XL2'00' if N/A                         *
*********************************************************************
SQLLPROG DS    0CL130             * Long Program
SQLLPRGL DS    H                  * Long Program Len
SQLLPRGT DS    CL128              * Long Program Name
SQLLCOL  DS    0CL130             * Long COLLID
SQLLCOLL DS    H                  * Long COLLID Len
SQLLCOLT DS    CL128              * Long COLLID
SQLLCURS DS    0CL130             * Long CURSOR Len
SQLLCURL DS    H                  * Long CURSOR Len
SQLLCURT DS    CL128              * Long CURSOR Name
SQLVARCL EQU   *-SQLLCURS         * Long Entity Length
         ORG   SQLLCOLL
SQLLSETC DS    0CL130             * IFCID 316 Long SET CURRENT
SQLLSETL DS    H                  * IFCID 316 Long SET CURRENT
SQLLSETT DS    CL128              * IFCID 316 Long SET CURRENT
SQLLQUAL DS    0CL130             * IFCID 316 Long Default Qualifier
SQLLQULL DS    H                  * IFCID 316 Long Default Qualifier
SQLLQULT DS    CL128              * IFCID 316 Long Default Qualifier
         ORG
SQLLONG# EQU   (*-SQLLPROG)/SQLVARCL * Number of entries
*
```

```
         DS    0F
*********************************************************************
*  SQLLOG Extension Record                                          *
*********************************************************************
         ORG   SQLLDATA
SQLERCDE DS    XL4                * SQL Text Len
SQLIRIDL DS    XL2                * RID List Flag
SQLEXTLN EQU   *-SQLLOG           * Extension Data Length
*********************************************************************
*  SQLLOG Extension Record Host Vars                                *
*********************************************************************
         ORG   SQLLDATA
SQLHVHDR DS    XL12               * HV Header
SQLHVDAT DS    CL5000             * HV Data
*
SQLHVLEN EQU   *-SQLHVHDR         * Maximum IFCID 247 Length
SQLIDSLN EQU   *-SQLLOG           * Maximum IFCID Data Length
*********************************************************************
*  SQL Text Record                                                  *
*********************************************************************
         ORG   SQLLDATA
SQLTXTLN DS    XL2                * SQL Text Len
SQLTEXTR DS    CL5000             * SQL Text in 5000 Byte chunks
         DS    CL256              * Filler
SQLTDSLN EQU   *-SQLLOG           * Maximum Text Data Length
```

A.3 Importing CA Detector® Data

This section describes the process to load CA Detector® data into an EZ-DB2 Workload.

This process assumes that you have already externalized the data from CA Detector®. Refer also to the CA Detector® documentation for information about externalizing the SQL data from CA Detector®.

In EZ-Tracer, create a new Workload and DRILL into the Trace Filters panel. In the following example, we have created a Trace workload called Detector.

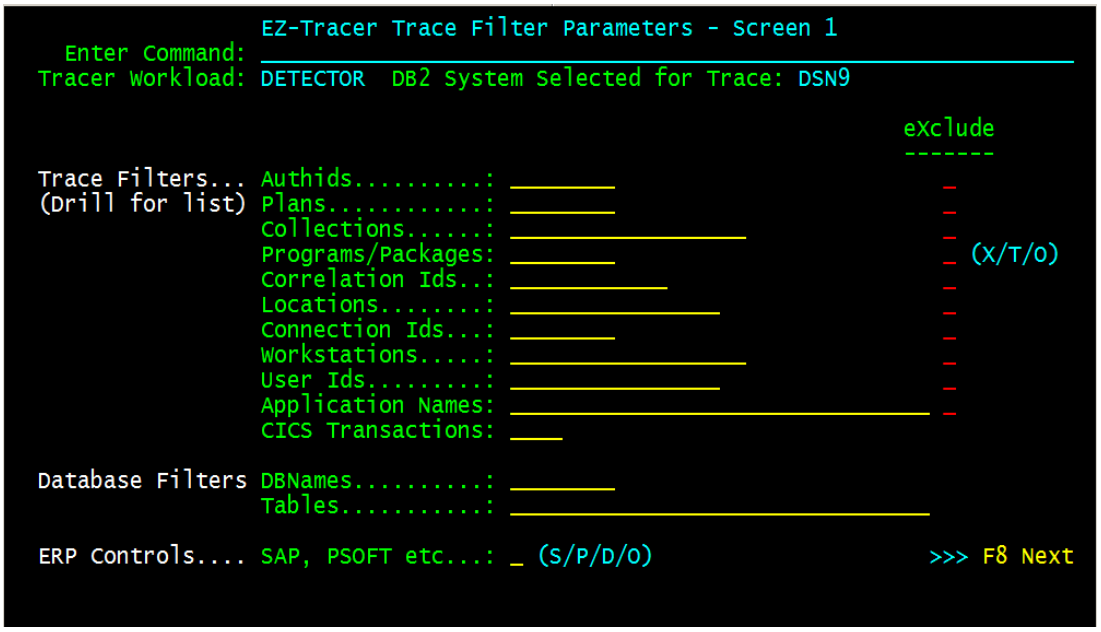
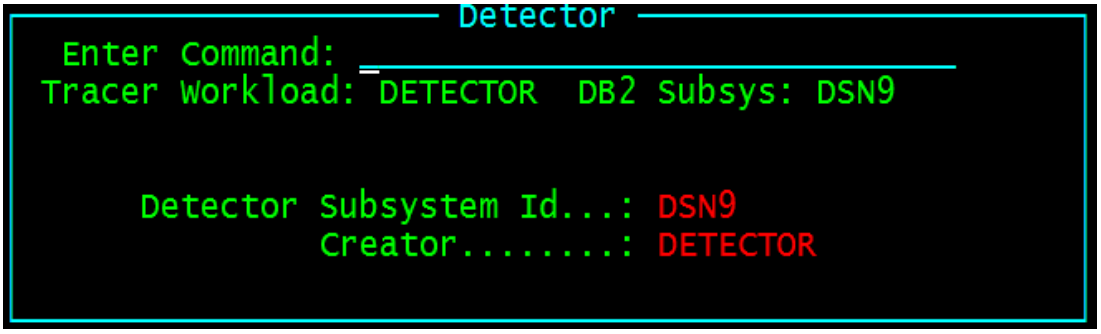


Figure 45 Import CA Detector Data

In the ERP Controls, enter **D** for Detector. The following Pop-Up window is displayed:



The following information is displayed on the panel:

Detector Subsystem ID — Specify the DB2 Subsystem ID where the CA Detector tables reside.

Creator — Specify the Creator used for the CA Detector® tables containing the externalized data. Refer to the CA Detector® documentation for information about externalizing the captured data.

Note: In the Trace Parameters, you can enter all the normal trace parameters, for example, FILTERS and SQL Consolidation Controls. Trace Time and Duration etc. are ignored. The DB2 Subs-system specified on the Start Trace panel should be the DB2 subsystem the SQL relates to. Refer to *the EZ-Tracer User Guide* for further information.

Once you have entered the data on this panel, continue to the normal tracer job submission and submit the job. The JCL contains the new step XOPLDET to load the CA Detector® data into the trace workload.

When the job has completed, you can view the Tracer workload using the EZ-Tracer reports menu in the normal way

You can also import the trace workload into an EZ-SQL Warehouse workload in the normal way, where it can be processed by EZ-Index Analyzer and /or EZ-XOP (Index Optimizer).

A.4 Add User DD statement to Tracer/Cache Job Stream

A user may have a requirement to include user specified DD statement(s) in the Tracer/cache job stream.

For example:

```
//HGBYPASS DD DUMMY
```

To activate this feature, add a member, containing the required DD card(s), named EXTRADD to *your.prefix*.CNTL

where *your.prefix* is the HLQ for the EZ-DB2 runtime CNTL library.

Any DD statements included in the member EXTRADD will be automatically inserted in every generated Tracer/Cache job stream.