

# **Dialog Trace Facility**

Small Programming Enhancement for  
CL/SUPERSESSION®

Version 147

G251239000

May 1998

Candle Corporation  
2425 Olympic Boulevard  
Santa Monica, California 90404

**Registered trademarks of Candle Corporation:** AF/OPERATOR, AF/PERFORMER, AF/REMOTE, Availability Command Center, Candle Command Center, Candle Electronic Customer Support, Candle Management Server, Candle Management Workstation, Candle Technologies, CL/CONFERENCE, CL/SUPERSESSION, CT, CT/Data Server, CT/DS, !DB, !DB/QUICKCHANGE, DELTAMON, OMEGACENTER, OMEGAMON, OMEGAMON/e, OMEGAMON II, OMEGAMON Monitoring Agents, OMEGAVIEW, OMEGAVIEW II, and Transplex.

**Trademarks of Candle Corporation:** Alert Adapter, Alert Emitter, AMS, Amsys, AUTOMATED FACILITIES, Availability Management Systems, CCC, CICAT, CL/ENGINE, CL/GATEWAY, CL/TECHNOLOGY, CMS, CMW, Command & Control, Connect-Notes, Connect-Two, CSA ANALYZER, CT/ALS, CT/Application Logic Services, CT/DCS, CT/Distributed Computing Services, CT/Engine, CT/Implementation Services, CT/IX, CT/Workbench, CT/Workstation Server, CT/WS, !DB/DASD, !DB/EXPLAIN, !DB/MIGRATOR, !DB/QUICKCOMPARE, !DB/SMU, !DB/Tools, !DB/WORKBENCH, Design Network, DEXAN, End-to-End, Enterprise Candle Command Center, Enterprise Candle Management Workstation, EPILOG, ESRA, HostBridge, IntelliWatch, Messaging Mastered, MQADMIN, MQEdit, MQEXPERT, MQMON, MQSecure, MQView, Networked Applications, Networked Businesses, OMC Gateway, OMC Status Manager, OMEGACENTER Bridge, OMEGACENTER Gateway, OMEGACENTER Status Manager, OMEGAMON Management Center, OSM, PC COMPANION, Performance Pac, PowerQ, PQConfiguration, PQEdit, PQScope, RTA, Solutions for Networked Applications, Solutions for Networked Businesses, Status Monitor, and Unified Directory Services.

**Service marks of Candle Corporation:** CECS.

Products and terms used herein may be trademarks or registered trademarks of their respective holders. This documentation set may contain references to some or all of the following products.

**Registered trademarks of International Business Machines Corporation:** AIX, Application System/400, AS/400, BookManager, DB2, IBM, NetView, OS/2, OS/400, SAA, Systems Application Architecture, SystemView, System Performance Monitor/2, THESEUS2, VM/ESA, and VTAM.

**Trademarks of International Business Machines Corporation:** ACF/VTAM, CICS, CICS/ESA, CICS/MVS, CICS/VSE, CICS/Transaction Server for OS/390 Release 1, Common User Access, CUA, DATABASE 2, DataHub, DFSMS, DFSMSdfp, DFSMSdss, DFSMSHsm, DFSMSrmm, ESA/370, ESCON, GDMM, Hiperspace, IIN, IMS/ESA, Information Warehouse, MQSeries, MVS/DFP, MVS/ESA, MVS/SP, MVS/XA, OfficeVision, OpenEdition, PR/SM, QMF, RACF, VM/XA, VSE/ESA, VSE/POWER, and 3090.

**Registered trademarks of Computer Associates International, Inc.:** CA-ACF2, CA-DATACOM, CA-DATACOM/DB, CA-IDEAL, CA-TOP SECRET, IDMS, and ROSCOE.

**Trademarks of Computer Associates International, Inc.:** CA-MAZDAMON, CA-7, and TMS.

**Registered trademarks of SAS Institute, Inc.:** SAS, SAS/GRAPH, and SAS/ETS.

**Trademarks and registered trademarks of other companies:** ADABAS is a trademark of Software/AG of North America, Inc. Advantis is a trademark of Advantis. ABEND-AID is a trademark of Compuware, Inc. DEC is a registered trademark of Digital Equipment Corporation. GENER/OL is a registered trademark of Pansophic Systems, Inc. HP-UX is a trademark of Hewlett-Packard Company. Key Tronic is a registered trademark of Key Tronic Corporation. Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation. Millennium is a registered trademark of Dun & Bradstreet Software Services, Inc. Multi-Image Manager and NetSpy are trademarks; and TPX is a registered trademark of Legent, Inc. Multiple Domain Facility and Amdahl are trademarks of Amdahl Corporation. MXG is a registered trademark of Merrill Consultants. NetWare and Novell are registered trademarks of Novell, Inc. ORACLE is a registered trademark of Oracle Corporation. RoadRunner and Maestro are trademarks of Unison Software. SAP is a registered trademark of SAP AG. Solaris and SunOS are trademarks of Sun Microsystems, Inc. SUPRA is a registered trademark of Cincom Systems, Inc. Tandem is a trademark of Tandem Computers, Inc. UFO is a registered trademark of On-Line Software International, Inc. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Ltd. USADirect is a registered trademark of AT&T. VMSECURE and SOLVE:Netmaster are registered trademarks of Sterling Software, Inc. Windows NT is a trademark; and Microsoft and Windows are registered trademarks of Microsoft Corporation.

All other trademarks and registered trademarks are the property of their respective companies.

ProtoView Development Corp. - Contains DataTable Version 3.0 Copyright 1989—1996 by ProtoView Development Corp. and distributed under license by Candle Corporation.

Copyright © 1993, 1998, Candle Corporation. All rights reserved. International rights secured.

NOTICE: This documentation is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions set forth in the applicable license agreement and/or the applicable government rights clause.

# Contents

---

<b>List of Illustrations</b> .....	<b>5</b>
<b>List of Tables</b> .....	<b>7</b>
<b>Read This First</b> .....	<b>9</b>
Preface .....	9
Customer Support .....	10
Documentation Conventions .....	12
Documentation Set .....	14
<b>Chapter 1. Introduction to the Dialog Trace Facility</b> .....	<b>17</b>
Overview .....	18
Types of Traces .....	19
Storing Trace Output .....	20
Resource Requirements .....	20
<b>Chapter 2. Preparing to Use DTF</b> .....	<b>23</b>
Getting Ready to Begin a Trace .....	24
Turning DTF On and Off .....	25
Traceable Compilation Mode .....	26
Security .....	27
Invoking DTF .....	28
<b>Chapter 3. Performing a Trace</b> .....	<b>29</b>
Overview .....	30
Enabling the User's Terminal .....	31
Selecting a Trace Type .....	33
Performing Flow Trace .....	34
Performing Statement Trace .....	37
Breakpoint Destination .....	44
Stopping a Trace .....	45
Maintaining Variables .....	48
Invoking the Trace Environment with an Empty Table .....	49
Using the Action Bar Selections .....	50
<b>Chapter 4. Putting It All Together</b> .....	<b>55</b>
Overview .....	56
Dialog Trace Scenario .....	57
<b>Chapter 5. Function and Command Reference</b> .....	<b>65</b>
Overview .....	66
DTRACE .....	67

REFRESH .....	72
REFRESH Function .....	74
<b>Chapter 6. Messages</b> .....	<b>75</b>
CL/ENGINE Messages .....	76
<b>Appendix A. Generalized Trace Facility Output</b> .....	<b>85</b>
Formatted Output .....	85
<b>Index</b> .....	<b>89</b>

## List of Illustrations

---

1. Dialog Trace Output — Trace View Panel (Flow Trace) . . . . . 36
2. Dialog Trace Output — Trace View Panel (Statement Trace) . . . . . 42
3. Dialog Trace Output — Trace View Panel (KLSZPTRK) . . . . . 58



## List of Tables

---

1. CL/SUPERSESSION and CL/GATEWAY Documentation . . . . .	14
2. Dialog Trace Output — Trace View Panel Actions (Flow Trace) . . . . .	36
3. Dialog Trace Output — Trace View Panel Actions (Statement Trace) . . . . .	43
4. GTF Output Keywords . . . . .	87





## Preface

---

### About this document

This document describes the small programming enhancement (SPE) known as the dialog trace facility (DTF). This guide explains how to use DTF in the CL/SUPERSESSION® environment to debug dialogs coded in Structured Session Procedure Language (SSPL).

This guide is designed for programmers who are familiar with CL/SUPERSESSION, CT/Engine™, and SSPL.

---

### Related documentation

Familiarity with the following Candle documents is recommended.

- *CL/SUPERSESSION Customization Guide*
- *CL/SUPERSESSION Dialog Language Reference Manual*
- *CL/SUPERSESSION Messages Manual*
- *CT/Engine SSPL Programming Guide*

# Customer Support

---

## Introduction

Candle provides electronic support and telephone support to assist you when you have questions about Candle products. Customer support is available 24 hours a day, seven days a week.

---

## Electronic support

Candle Electronic Customer Support® (CECS<sup>SM</sup>) enables you to search for existing questions, answers, problems, and fixes; review preventive service planning (PSP) information; and open incidents for Candle products. CECS is available through the Advantis<sup>TM</sup> network and by direct PC dial-up. For registration information, call your nearest Candle Support Services office.

---

## Telephone support

If you have an urgent problem or need to talk to a Candle Support Services representative, contact the Support Services office nearest you.

Office	Telephone	Fax
<b>North America</b>		
Santa Monica	(800) 328-1811 (310) 829-5844	(310) 582-4204
<b>Europe</b>		
Antwerp	(32) 3 270 95 60	(32) 3 270 95 41
Breda	(31) 76 533 35 50	(31) 76 533 35 10
Manchester	(44) 161 437 5224	(44) 161 437 5225
Munich	(49) 89 54554 222	(49) 89 54554 119
Paris	(33) (1) 5361 60 60	(33) (1) 5361 06 16
Sollentuna	(46) 8 623 1235	(46) 8 623 1855
<b>Asia Pacific</b>		
Hong Kong	(852) 2528 6289	(852) 2865 0770
Kuala Lumpur	(603) 230 9930	(603) 230 9932
Singapore	(65) 220 50 92	(65) 226 35 79
Sydney	(61) 2 9954 1500	(61) 2 9954 1818
Tokyo	(81) 3 5562-6991	(81) 3 5562-6995

---

## International customers

When your local support office is unavailable, you may contact Candle's North America support center. If USADirect® service is available in your country, use the 800 telephone number. If USADirect service is not available, ask your international operator for assistance in calling Candle's local (310) number.

---

## Incident information

A Candle support representative may request the following information when you call to report a problem:

- your Candle personal ID (PID) number
- the release level of the Candle product
- the release level of IBM® or other vendor software
- identifying information and dates of recently applied maintenance to the Candle product
- a detailed description of the problem (including the error message) and what led up to the failure
- a description of any unusual events that occurred before the problem

---

## Incident documentation

You may be asked to send incident documentation to Candle Support Services. On the outside of all packages you send, please write the incident number given to you by the Candle Support Services representative.

Send tapes containing the incident information to the following address, unless directed otherwise by your Candle Support Services representative:

**Candle Support Team**  
**Candle Support Center, *Incident number***  
**2425 Olympic Boulevard**  
**Santa Monica, CA 90404**

Send all other documentation, for example, diskettes or paper documentation, such as screen prints, to the address that will be given to you by your Candle Support Services representative.

# Documentation Conventions

---

## Introduction

Candle documentation adheres to accepted typographical conventions for command syntax. Conventions specific to Candle documentation are discussed in the following sections.

---

## Panels and figures

The panels and figures in this document are representations. Actual product panels may differ.

---

## Revision bars

Revision bars (|) may appear in the left margin to identify new or updated material.

---

## Variables and literals

In examples of command syntax, uppercase letters are actual values (literals) that the user should type; lowercase letters are used for variables that represent data supplied by the user. Default values are underscored.

**LOGON APPLID(ccccccc)**

In the above example, you type **LOGON APPLID** followed by an application identifier (represented by *ccccccc*) within parentheses. The application identifier can have at most eight characters.

**Note:** In ordinary text, variable names appear in italics.

---

## Symbols

The following symbols may appear in command syntax.

Symbol	Usage
	<p>The 'or' symbol is used to denote a choice. Either the argument on the left or the argument on the right may be used. Example:</p> <p><b>YES   NO</b></p> <p>In this example, YES or NO may be specified.</p>
[ ]	<p>Denotes optional arguments. Those arguments not enclosed in square brackets are required. Example:</p> <p><b>APPLDEST DEST [ALTDEST]</b></p> <p>In this example, DEST is a required argument and ALTDEST is optional.</p>
{ }	<p>Some documents use braces to denote required arguments, or to group arguments for clarity. Example:</p> <p><b>COMPARE {workload} - REPORT={SUMMARY   HISTOGRAM}</b></p> <p>The <i>workload</i> variable is required. The REPORT keyword must be specified with a value of SUMMARY or HISTOGRAM.</p>
-	<p>Default values are underscored. Example:</p> <p><b>COPY infile outfile - [COMPRESS={<u>YES</u>   NO}]</b></p> <p>In this example, the COMPRESS keyword is optional. If specified, the only valid values are YES or NO. If omitted, the default is YES.</p>
␣	<p>The symbol ␣ indicates a blank space, when needed for clarity.</p>

## Documentation Set

---

### Introduction

Candle provides a complete set of documentation for CL/SUPERSESSSION and CL/GATEWAY. Each manual in this documentation set contains a specific type of information to help you use the product.

Candle welcomes your comments and suggestions for changes or additions to the documentation set. A user comment form, located at the back of each manual, provides simple instructions for communicating with Candle's Information Development department.

---

### Product documentation

The documentation listed in the following table is available for CL/SUPERSESSSION and CL/GATEWAY. To order additional product manuals, contact your Candle Support Services representative.

<b>Document Number</b>	<b>Document Name</b>	<b>Description</b>
LS60-3779	Version 147 Release Guide	Contains new information for this release.
LS99-3783	Program Directory	Provides installation instructions and details all other installation considerations.
LS55-3785	Basic Configuration Guide	Provides basic instructions for customizing CL/SUPERSESSSION and CL/GATEWAY to the specific needs of your network, system, and users.
LS51-3781	Customization Guide	Provides instructions and explanations for customizing CL/SUPERSESSSION and CL/GATEWAY to the needs of your network, system, and users.
LS54-3786	User's Guide	Contains brief instructions on how to operate CL/SUPERSESSSION and CL/GATEWAY.
LS99-3789	Operator's Guide	Describes the CT/Engine operator facility and commands used by CT/Engine, CL/SUPERSESSSION, and CL/GATEWAY.

<b>Table 1 (Page 2 of 2). CL/SUPERSESSION and CL/GATEWAY Documentation</b>		
<b>Document Number</b>	<b>Document Name</b>	<b>Description</b>
LS99-4225	Introduction to SSPL Dialogs	Introduces users to the Structured Session Procedure Language (SSPL); shows how to customize and use some simple dialogs written in SSPL.
LS99-3821	SSPL Programming Guide	Explains how to create your own dialogs with SSPL, using a sample application that creates and manages a table.
LS53-3787	Dialog Language Reference Manual	Contains comprehensive descriptions of all features of the SSPL dialog language.
LS57-3780	Problem Determination Guide	Contains instructions and documentation recommendations for locating and solving problems in CL products.
LS52-3788	Messages Manual	Lists and explains all CT/Engine, CL/SUPERSESSION, and CL/GATEWAY messages and suggests appropriate user actions.
LVM99-4103	Quick Reference Card	Pocket-sized document that contains step-by-step instructions for using CL/SUPERSESSION and CL/GATEWAY.
LS59-3801	Master Index	Contains a master index for all CL/SUPERSESSION and CL/GATEWAY manuals that contain indexes.





# Chapter 1. Introduction to the Dialog Trace Facility

---

Overview .....	18
Types of Traces .....	19
Storing Trace Output .....	20
Resource Requirements .....	20

## Overview

---

### Definition

The dialog trace facility (DTF), a component of CT/Engine's dialog manager, is an interactive, panel-driven tool for tracing dialogs written in Structured Session Procedure Language (SSPL).

---

### Advantages

With various tracing options and an easy-to-learn interface, the dialog trace facility can be of great assistance in testing and debugging SSPL dialogs.

With DTF, you can

- trace the flow of control between dialogs and applications
- trace a group of statements within a dialog
- store trace output in a table
- store trace output in the General Trace Facility (GTF).

These tracing features are described in the sections that follow.

---

### Implementation

To implement this SPE, apply PTFs QLV1109 and QLV1215. The dialog trace facility is presented only in English.

# Types of Traces

---

## Introduction

You can use DTF to generate two types of trace records: flow trace and statement trace.

---

## Flow trace

Flow trace provides an overview of dialogs, generating trace entries that show the transfer of control between dialogs. Once you activate flow trace, all dialogs that are executed on a specific LU are traced.

---

## Statement trace

Statement trace provides detailed information about statements executing within a dialog. You can specify a display of all executable statements or a selected range of statements from a dialog.

The breakpoint processing option of statement trace allows you to stop the execution of dialogs at a specific statement. At the breakpoint, you can check the value of variables, modify variables, and change the current trace environment *before* the statement is executed. You can specify any number of breakpoints in a dialog.

## Storing Trace Output

---

### User trace table

Select the user trace table as your output destination when you want to view trace information as soon as tracing completes. The user trace table displays the dialogs or statements that you have selected for tracing immediately after they are executed. (Figure 1 on page 36 is an example of a user trace table.)

**Breakpoint processing:** Trace table storage allows interactive tracing, via breakpoint processing. Breakpoint processing stops an executing dialog at each breakpoint. This enables you to check dialog flow and dialog statement execution and to modify variables before the dialog completes execution.

This output destination is not recommended when you plan to generate a great deal of trace data. When the table is filled, DTF begins overlaying existing table entries with newer ones (FIFO).

---

### Generalized trace facility

You can also use the generalized trace facility (GTF) as your output destination. Use GTF when you plan to generate large quantities of trace records. You can print the output and thus have all data at hand at one time.

Interactive tracing is not available when GTF output is specified.

For examples of GTF output, see “Generalized Trace Facility Output” on page 85.

---

## Resource Requirements

---

### Caution

CPU and storage usage due to DTF can be significant. Therefore, Candle recommends that the system administrator limit the number of DTF users.

---

### Compiled dialog storage

Statement tracing results in a 15% increase (approximately) in dialog object module size. You must ensure that you have adequate storage space for the statements from dialogs that are compiled for statement trace. To review current storage allocation, issue the STORAGE DETAIL command from the CUA™ operator.

---

The table below contains the storage estimates for CL/SUPERSESSSION and CT/Engine. These storage estimates are calculated for traceable compilation of all CL/SUPERSESSSION and CT/Engine dialogs. Selective compilation, which is the recommended option, uses less storage.

<b>Component</b>	<b>Number of Members</b>	<b>Number of Statements</b>	<b>Storage Amount</b>
CL/SUPERSESSSION	458	13K	460 KB
CT/Engine	377	20K	337 KB

Compiled dialog storage is obtained from contiguous, page-isolated storage, so that it does not contribute to the working set size when DTF is not on. It is allocated in 100 KB blocks.

---

### **Table storage**

Several temporary, nonshared tables are used for interactive tracing. These tables require approximately 2 to 8 KB of storage per interactive trace user when you use the default user trace table size (255). More table storage is used if a larger user trace table size is specified when enabling an LU for trace.

---

### **Range storage**

Range storage is required whenever ranges are declared for users. The requirement is approximately 0.5 to 1 KB per enabled user.

Range storage is obtained from a contiguous, page-isolated pool common to all users so that it does not contribute to the working set when statements are not being traced (for example, when enabled users are not invoking dialogs with declared ranges).



## Chapter 2. Preparing to Use DTF

---

Getting Ready to Begin a Trace .....	24
Turning DTF On and Off .....	25
Traceable Compilation Mode .....	26
Security .....	27
Invoking DTF .....	28

## Getting Ready to Begin a Trace

---

### Introduction

Before you actually begin to trace dialogs, you must be authorized and the system must be prepared. These actions often require the services of the operator and system administrator.

---

### Procedure

The following table describes the steps to be performed, who performs them, and where instructions for performing them can be found in this document.

Step	Action	Explained in . . .
1	Have the operator turn on DTF (if it is not already on).	“Turning DTF On and Off” on page 25
2	Invoke DTF.	“Invoking DTF” on page 28
3	Enable your session for tracing either by having the operator enable your session or by executing the trace dialog (DIALOG KLVDTRC).	“Overview” on page 30

When you have completed these steps (which are described in the specified sections), you are ready to perform a trace. (See “Performing a Trace” on page 29.)



# Turning DTF On and Off

---

## Introduction

Before a user can turn on DTF, the system administrator must give administrator authority to that user's ID.

---

## Important

Users enabled for DTF can circumvent normal security procedures by altering dialog flow. Therefore, Candle recommends that the system administrator is the only person who authorizes users to turn on DTF.

---

## Turning on DTF

To turn on DTF, use the following procedure.

Step	Action
1	Select <b>Trace</b> on the CT/Engine operator facility action bar. <i>Result:</i> The Trace pull-down menu appears.
2	Select <b>Dialog Trace</b> . <i>Result:</i> The DTRACE Commands pop-up appears.
3	Select <b>Turn Dialog Trace On</b> . <i>Result:</i> The CT/Engine Operator panel displays the confirming log message: <b>KLVOP416 DIALOG TRACE IS ON</b> .

---

## Turning off DTF

Use the following procedure to turn off DTF.

Step	Action
1	Select <b>Trace</b> from the CT/Engine operator facility action bar. <i>Result:</i> The Trace pull-down menu appears.
2	Select <b>Dialog Trace</b> . <i>Result:</i> The DTRACE Commands pop-up appears.
3	Select <b>Turn Dialog Trace Off</b> . <i>Result:</i> The CT/Engine Operator panel displays the confirming log message: <b>KLVOP416 DIALOG TRACE IS OFF</b> .

# Traceable Compilation Mode

---

## Introduction

Traceable compilation mode is a selection that you can make on the DTRACE Commands pop-up. While this mode is turned on, each time a dialog is refreshed it is compiled with object modules that can be traced.

---

## Important

Use of traceable compilation mode can significantly increase overhead. Candle recommends that you instead use the REFRESH command to recompile a specific dialog that you want to trace.

---

## Procedure to start

To activate traceable compilation mode, use the following procedure.

Step	Action
1	Select <b>Trace</b> on the CT/Engine operator facility action bar. <i>Result:</i> The Trace pull-down menu appears.
2	Select <b>Dialog Trace</b> . <i>Result:</i> The DTRACE Commands pop-up appears.
3	Select <b>Start Traceable Compilation Mode</b> . <i>Result:</i> The CT/Engine Operator panel displays a message confirming that traceable compilation mode is on.

---

## Procedure to end

Use the following procedure to turn off traceable compilation mode.

Step	Action
1	Select <b>Trace</b> from the CT/Engine operator facility action bar. <i>Result:</i> The Trace pull-down menu appears.
2	Select <b>Dialog Trace</b> . <i>Result:</i> The DTRACE Commands pop-up appears.
3	Select <b>End Traceable Compilation Mode</b> . <i>Result:</i> The CT/Engine Operator panel displays a message confirming that traceable compilation mode is off.

---

## Security

---

### Caution

Users enabled for DTF have the ability to refresh dialogs, run dialogs from the DTF panels, and alter dialog flow. Altering dialog flow can circumvent normal security procedures at your site.

Because of this security circumvention, enabling users for DTF should be controlled by the system administrator.

# Invoking DTF

---

## Introduction

You can invoke DTF with the dialog KLVDTRC using the following procedure.

---

## On the command line

Invoke DTF with the dialog KLVDTRC using the following procedure.

**Note:** You must have administrator authority to issue DIALOG KLVDTRC.

- ▶ Type **DIALOG KLVDTRC** on the CL/SUPERSESSION Main Menu command line and press Enter.

**Result:** The first panel to appear depends on several factors. See the table under “Next step” on page 30 to determine your next step.

If you receive the message **DIALOG TRACE IS NOT TURNED ON**, ask your product administrator or system operator to turn on DTF, then invoke dialog trace again.

## Chapter 3. Performing a Trace

---

Overview .....	30
Enabling the User's Terminal .....	31
Selecting a Trace Type .....	33
Performing Flow Trace .....	34
Performing Statement Trace .....	37
Breakpoint Destination .....	44
Stopping a Trace .....	45
Maintaining Variables .....	48
Invoking the Trace Environment with an Empty Table .....	49
Using the Action Bar Selections .....	50

# Overview

---

## Introduction

Now that you have invoked DTF, you are ready to start a trace. But first you must establish the trace environment and specify the DTF features you want to use.

---

## Required authority

You must be an administrator or be authorized by an administrator to perform a trace.

---

## Next step

After invoking DTF, your next step depends on whether or not you have already enabled your terminal and established the trace environment. Use the following table as a guide.

<b>IF . . .</b>	<b>THEN . . .</b>	<b>EXPLAINED IN . . .</b>
the Dialog Trace Enable Options pop-up appears	you must enable yourself for DTF	“Enabling the User's Terminal” on page 31
the Dialog Trace pop-up appears	you can now select a trace option	“Selecting a Trace Type” on page 33
the Dialog Trace Output — Trace View panel appears	you can begin to analyze trace data	One of the following:  <b>Flow trace</b> “Performing Flow Trace” on page 34  <b>Statement trace</b> “Performing Statement Trace” on page 37

# Enabling the User's Terminal

---

## Introduction

If your terminal has not yet been enabled for DTF, perform the following procedure.

---

## Procedure

1. Complete the following fields on the Dialog Trace Enable Options pop-up.

Field	Description
Trace entries	<p>The maximum number of entries to be stored in the user trace table. The default is 255.</p> <p>The expected number of entries varies depending on the type of trace you are running:</p> <p><b>Flow trace</b> Consider how many dialogs you want to display as output.</p> <p><b>Statement trace</b> Consider the size of the test dialog and the number of ranges you want to examine.</p>
Keep trace environment	<p>Specifies whether or not to retain the trace environment when you log off. The default is YES.</p> <p>If you will be logging on and off during testing, specify YES so you will not have to re-establish the trace environment each time.</p>

2. When you have completed your entries, press Enter.  
**Result:** The trace environment is now established.
3. Proceed as instructed in the following table.

<b>IF . . .</b>	<b>THEN . . .</b>
the user trace table contains no entries	the Dialog Trace pop-up appears. Proceed to “Selecting a Trace Type” on page 33.
the user trace table does contain entries	<p>the Trace View Panel appears. Proceed to one of the following:</p> <p><b>Flow trace</b>  “Performing Flow Trace” on page 34</p> <p><b>Statement trace</b>  “Performing Statement Trace” on page 37</p>



## Selecting a Trace Type

---

### Introduction

If your user trace table contains no entries, the Dialog Trace pop-up appears, enabling you to select the kind of trace that you want to run.

---

### Procedure

1. Select an option from the Dialog Trace pop-up.
2. Follow the appropriate procedure. (The table below directs you to helpful information.)

<b>IF ...</b>	<b>THEN ...</b>
you selected <b>Flow trace processing</b>	proceed to “Performing Flow Trace” on page 34
you selected <b>Statement trace processing</b>	proceed to “Performing Statement Trace” on page 37
you selected <b>Refresh a dialog</b>	proceed to “Refreshing a dialog (using pop-up)” on page 37
you selected <b>Invoke trace environment with empty table</b>	proceed to “Invoking the Trace Environment with an Empty Table” on page 49

# Performing Flow Trace

---

## Introduction

If you selected the flow trace option on the previous panel, the Set Flow Destinations pop-up appears. This panel allows you to start a flow trace.

---

## Procedure

To perform a flow trace, do the following steps.

Step	Action
1	<p>Fill in the Set Flow Destinations fields using the following guidelines:</p> <p><b>Table destination</b> Specifies that output is written to the user trace table. <b>Note:</b> This destination is invalid if the trace environment was established with TBLSIZE=0.</p> <p><b>Breakpoint destination</b> Specifies dynamic invocation of the breakpoint dialog when one dialog transfers control to another dialog. When Breakpoint is On, output is written to the user trace table. See “Breakpoint Destination” on page 44 for more information. <b>Note:</b> This destination is invalid if the trace environment was established with TBLSIZE=0.</p> <p><b>GTF destination</b> Specifies that output is written to GTF. <b>Note:</b> Before trace records can be written to GTF, GTF must be started and the GTF operator command issued.</p>
2	<p>Press Enter after making your selection.</p> <p><b>Result:</b> A message displays, confirming flow trace processing and the destination type you selected.</p>
3	<p>Press F3 to return to the application in which the test dialog runs, and execute the dialog.</p>

---

## Viewing trace data

Your access to the information generated by the trace is determined by your chosen destination.

<b>IF . . .</b>	<b>THEN . . .</b>
you specified <b>Table destination</b>	after the traced dialog has completed, invoke the DTF dialog again (using one of the methods listed in “Invoking DTF” on page 28)  <i>Result:</i> The Dialog Trace Output — Trace View panel, shown in Figure 1 on page 36, is displayed.
you specified <b>Breakpoint destination</b>	the DTF dialog is invoked automatically while the dialog is being traced  <i>Result:</i> The Dialog Trace Output — Trace View panel, shown in Figure 1 on page 36, is displayed. Press F3 to proceed to the next breakpoint. (If you have already reached the last breakpoint, you exit from the panel when you press F3.)
you specified <b>GTF destination</b>	you stay in your application while tracing continues; later, use GTF to access trace data

---

## User trace table

The Dialog Trace Output — Trace View panel displays the user trace table. This panel shows the flow of control, indicating calls and SYSPARM information for each dialog. A scroll indicator in the upper right area of the panel indicates line and column location.

```

_____ Actions Environment View Utility Options Help
----- 06/04/93 15:03:44
                        Dialog Trace Output – Trace View

Enter an action or select an Action Bar item, then press Enter.
S=Source list  R=Range update

Dialog  Mem/Copy  Stmt Data                                Lines _1 to 16 of 16
                                           Columns _1 to 53 of 87
-----
- SCMD                                INVOKED SMSGEN , SYSPARM= SMSGLCOK
- SMSGEN                              INVOKED SMSGLEN , SYSPARM= SMSGLCOK
- SMSGLEN                             RETURNED TO SMSGEN , RC=
- SMSGEN                              INVOKED SMSGBLD , SYSPARM= CL001,W,L,Request successf
- SMSGBLD                             RETURNED TO SMSGEN , RC=
- SMSGEN                              RETURNED TO SCMD , RC=
- SCMD                                RETURNED TO SVSELEN , RC= 1
- SVSELEN                             INVOKED SFKEYS2 , SYSPARM= Enter,F1=Help,F3=Exit,F5=R
- SFKEYS2                             RETURNED TO SVSELEN , RC=
- SVSELEN                             INVOKED SSUNIN , SYSPARM=
- SSUNIN                              RETURNED TO SVSELEN , RC=
- SVSELEN                             INVOKED SSINITEN, SYSPARM=
- SSINITEN                            RETURNED TO SVSELEN , RC=
- SVSELEN                             INVOKED SVKEYS , SYSPARM= Enter F1=Help F3=Exit F
- SVKEYS                              RETURNED TO SVSELEN , RC= 0
- SVSELEN                             INVOKED SCMD1 , SYSPARM=
***** BOTTOM OF DATA *****

F1=Help F2=Keys F3=Exit F7=Bkwd F8=Fwd F10=Action Bar

```

**Figure 1. Dialog Trace Output — Trace View Panel (Flow Trace)**

## Panel actions

For more information about a particular dialog, enter an action code next to the dialog name. The following actions are available.

<b>Action</b>	<b>Result</b>
Source list (S)	Lists dialog source statements for dialogs that have been compiled for trace.  If you select a dialog that has not been compiled for trace, no data will be displayed. If this happens, refresh the dialog for traceable compilation and try again. “Refreshing a dialog (using pop-up)” on page 37 contains instructions for dialog refresh.
Range update (R)	Displays a currently defined range of statements. You can then add and delete ranges for trace.

# Performing Statement Trace

---

## Introduction

In order to perform a statement trace on a dialog, you must perform the following procedures:

- Refresh the dialog, that is, recompile the dialog in traceable compilation mode.
- Specify the trace, that is, define a dialog statement or range of statements for tracing.

These procedures are described in the following sections.

---

## Refreshing a dialog (using pop-up)

To refresh a dialog, perform the following steps.

Step	Action
1	Select <b>Refresh a dialog</b> from the Dialog Trace pop-up. <i>Result:</i> The REFRESH Dialog pop-up appears.
2	Fill in the fields using the following guidelines: <b>Member Name</b> Specify the dialog you want to refresh. <b>Compile mode</b> Specify the compilation mode for the refresh. <b>TRACE</b> Traceable compilation mode. The dialog is compiled to enable tracing and produces output that can be routed to a table or GTF. <b>NOTRACE</b> Nontraceable compilation mode. The dialog is compiled normally and with no trace output available.
3	Press Enter when you have finished filling in the panel. <i>Result:</i> A message confirming that the dialog has been refreshed displays. If the dialog does not refresh, review the output in the view log for error messages.

---

## Refreshing dialogs (using a CLIST)

Instead of manually issuing an individual REFRESH command for each dialog, you can create a CLIST member in TLVCMD5 to issue the REFRESH commands for all the dialogs that you are testing.

Step	Action
1	In &lvddcmd, create a new member. Remember to begin the member name with a dollar sign (\$).  <b>Note:</b> Do not name your CLIST member \$REFRESH (or <i>dollar sign</i> followed by the name of any other CT/Engine command). In such a case, when you try to call up the CLIST member (by issuing the name <i>without the dollar sign</i> ), CT/Engine interprets and executes REFRESH as the command, not the member name.
2	On the first line, type <b>REFRESH P name</b> where <i>name</i> is the name of the dialog that you want to refresh. Repeat this procedure (on separate lines) for each dialog that you want to refresh.
3	Press F3.  <b>Result:</b> You exit from the member and your entries are saved.
4	On the CT/Engine operator panel command line, type the name of the CLIST member without the dollar sign and press Enter.  <b>Result:</b> The dialogs are refreshed in their order of listing in the CLIST member. A confirming message appears on the CT/Engine operator panel as each dialog is refreshed.



Step	Action
5	<p>Select one of the following statement trace destinations:</p> <p><b>Table destination</b> Specifies that output is written to the user trace table. <b>Note:</b> This destination is invalid if the trace environment was established with TBLSIZE=0.</p> <p><b>Breakpoint destination</b> Specifies dynamic invocation of the breakpoint dialog when one dialog statement transfers control to another dialog statement. When Breakpoint is On, output is written to the user trace table. See “Breakpoint Destination” on page 44 for more information. <b>Note:</b> This destination is invalid if the trace environment was established with TBLSIZE=0.</p> <p><b>GTF destination</b> Specifies that output is written to GTF. <b>Note:</b> Before trace records can be written to GTF, GTF must be started and the GTF operator command issued.</p>
6	<p>Press Enter.</p> <p><b>Result:</b> The Trace Ranges pop-up appears with the range and destination for the specified dialog displayed.</p>
7	<p>Press F12.</p> <p><b>Result:</b> The Source Statement List appears.</p>



Step	Action
8	<p>(Optional) Select any of the following actions to specify the table characteristics.</p> <p><b>Range Update</b> Displays a list of the currently defined ranges. You can then add and delete ranges for trace.</p> <p><b>Variable Update</b> Displays a list of the currently defined variables. You can then add and delete variables for trace.</p> <p><b>Variable Add</b> Defines a variable. (This selection is a fast path for the Variable Update and Add selection combination.)</p> <p>Variables are discussed in detail in “Maintaining Variables” on page 48.</p>

---

### Running the trace

To trace the dialog statements, perform the following steps:

Step	Action
1	From the Source Statement List pop-up, return to the application in which the test dialog runs.
2	Execute the test dialog.

---

### Viewing trace data

Your access to the information generated by the trace is determined by your chosen destination.

IF ...	THEN ...
you specified <b>Table destination</b>	<p>after the traced dialog has completed, invoke the DTF dialog again (using one of the methods listed in “Invoking DTF” on page 28)</p> <p><b>Result:</b> The Dialog Trace Output — Trace View panel, shown in Figure 2 on page 42, is displayed.</p>

IF ...	THEN ...
you specified <b>Breakpoint destination</b>	the DTF dialog is invoked automatically while the dialog is being traced  <b>Result:</b> The Dialog Trace Output — Trace View panel, shown in Figure 2 on page 42, is displayed.
you specified <b>GTF destination</b>	you stay in your application while tracing continues; later, use GTF to access trace data

---

## User trace table

The Dialog Trace Output — Trace View panel displays the user trace table. This panel shows the dialog, the dialog member, the statement number, and the statement. A scroll indicator in the upper right area of the panel indicates line and column location.

```

_____ Actions Environment View Utility Options Help
----- 06/04/93 15:03:44
                Dialog Trace Output - Trace View

Enter an action or select an Action Bar item, then press Enter.
  S=Source list  R=Range update
                Lines  1 to 16 of 16
Dialog  Mem/Copy  Stmt Data                Columns  1 to 53 of 87
-----
_ KLVDTTST KLVDTTST   32 set testzz 'Shared variable'
***** BOTTOM OF DATA *****

Breakpoint encountered in KLVDTTST at statement 32
F1=Help F2=Keys F3=Exit F7=Bkwd F8=Fwd F10=Action Bar

```

**Figure 2. Dialog Trace Output — Trace View Panel (Statement Trace)**

---

## Panel actions

For more information about a particular dialog, enter an action code next to the dialog name. The following actions are available.

<b>Table 3. Dialog Trace Output — Trace View Panel Actions (Statement Trace)</b>	
<b>Action</b>	<b>Result</b>
Source list (S)	<p>Lists dialog source statements for dialogs that have been compiled for trace.</p> <p>If you select a dialog that has not been compiled for trace, no data will be displayed. If this happens, refresh the dialog for traceable compilation and try again. “Refreshing a dialog (using pop-up)” on page 37 contains instructions for dialog refresh.</p>
Range update (R)	Displays a currently defined range of statements. You can then add and delete ranges for trace.

## Breakpoint Destination

---

### Description

Breakpoints are present in the code of some dialog statements. If you select **Breakpoint destination**, execution of a dialog is stopped each time a statement that contains a breakpoint is encountered.

---

### Breakpoint process

The table below describes the process that allows you to view each breakpoint and make changes as required.

Step	Agent	Action
1	User	Executes a dialog or dialog statement. <i>Result:</i> Control is transferred from one dialog to another or from one dialog statement to another.
2	DTF	Invokes the DTF dialog. <i>Result:</i> The Dialog Trace Output — Trace View panel (Figure 1 on page 36) is displayed.
3	User	Presses F3 repeatedly. <i>Result:</i> Each time the user presses F3, the Dialog Trace Output — Trace View panel reappears with a new entry.

If, at some point during statement processing, you want to make a change (for instance, modifying a variable, as explained in “Maintaining Variables” on page 48), do so, then begin the process again at step 1.

# Stopping a Trace

---

## Introduction

You may decide to stop a trace before the traced dialog has completed. For example, if the trace table is full you may wish to prevent DTF from overwriting the older entries.

There are several ways to suspend or stop tracing. The one you choose depends on the desired consequences:

### **Suspend trace output**

Trace environment remains active but no entries are added to the user trace table

### **Turn off flow trace**

The trace environment is terminated.

### **Bypass breakpoint**

Trace environment remains active but any remaining breakpoints for that specific dialog are bypassed during execution. Thus no entries are added to the user trace table for that specific dialog.

**Delete range** The trace environment is terminated.

---

## Suspending trace output

Use the following procedure to suspend trace output:

<b>Step</b>	<b>Action</b>
1	Select <b>Suspend trace output</b> on the Options pull-down. <b>Result:</b> The trace environment remains active but no new entries are added to the user trace table.
2	Press F3. <b>Result:</b> You exit the Dialog Trace Output — Trace View panel and return to the CL/SUPERSESSION Main Menu.

---

## Turning off flow trace

Use the following procedure to turn off flow trace:

Step	Action
1	Select <b>Flow trace</b> on the Environment pull-down. <i>Result:</i> The Set Flow Destinations pop-up appears.
2	Turn off the flow trace destination. <i>Result:</i> The trace environment is deactivated.
3	Press F3. <i>Result:</i> You exit the Dialog Trace Output — Trace View panel and return to the CL/SUPERSESSION Main Menu.

---

## Bypassing breakpoint

Use the following procedure to bypass the currently executing dialog's remaining breakpoints:

Step	Action
1	Select <b>Bypass breakpoint once</b> on the Options pull-down. <i>Result:</i> The trace environment remains active but no entry is added to the user trace table for that specific dialog.
2	Press F3. <i>Result:</i> You exit the Dialog Trace Output — Trace View panel and return to the CL/SUPERSESSION Main Menu.

---

## Deleting ranges

Use the following procedure to delete the statement range of a currently executing dialog:

Step	Action
1	Type <b>R</b> to the left of any dialog statement on the Dialog Trace Output — Trace View panel and press Enter. <i>Result:</i> The Trace Ranges pop-up appears listing all refreshed dialogs and their ranges.

Step	Action
2	Type <b>D</b> to the left of the current dialog (that is, the dialog currently being processed on the Dialog Trace Output — Trace View panel) and press Enter. <i>Result:</i> The Delete Confirmation pop-up appears.
3	Select <b>Delete range entry</b> and press Enter. <i>Result:</i> The dialog is deleted from the Trace Ranges pop-up.
4	Press F12. <i>Result:</i> You return to the Dialog Trace Output — Trace View panel.
5	Press F3. <i>Result:</i> You exit the Dialog Trace Output — Trace View panel and return to the CL/SUPERSESSSION Main Menu.

---

### More information

See “Using the Action Bar Selections” on page 50 for more information on the Options and Environment action bar choices.

# Maintaining Variables

## Introduction

Once you have generated output through statement trace processing, you can add, delete, or modify the value of variables within executing statements.

## Procedure

Perform the following procedure to add, delete, or modify variables.

Step	Action
1	<p>On the Dialog Trace Output — Trace View panel, type <b>V</b> next to a dialog name and press Enter.</p> <p><b>Result:</b> The Trace Variables pop-up displays the current variables and their values.</p> <p><b>Note:</b> If no variables have been previously defined, a placeholder row is displayed.</p> <pre> +-----+   Trace Variables for LU123456  Enter an action for the variable, then press Enter.                            A=Add Variable D=Delete Variable M=Modify Variable  Variable Scope   Dialog   Value                                     Lines 1 to 1 of 1    ----- ----- ----- -----    TESTUSER Session                USER01                                    *-----* BOTTOM OF DATA *-----*   F1=Help F2=Keys F7=Bkwd F8=Fwd F12=Cancel   +-----+ </pre>
2	<p>Take any of the following actions:</p> <p><b>Add Variable (A)</b> Adds a variable to a list that can be checked during tracing to determine the value.</p> <p><b>Delete Variable (D)</b> Deletes a variable from a list.</p> <p><b>Modify Variable (M)</b> Modifies the value of a variable during the trace process.</p>



# Invoking the Trace Environment with an Empty Table

---

## Introduction

Use this option if you want to display an empty user trace table. Then, before you begin a trace, you can perform preparatory actions. For example, you can

- execute a dialog via the Utility pull-down to gain diagnostic information that dialog trace may not provide
- do pretrace processing (for example, setting up conditions and/or tables) before performing tracing
- define ranges via the Environment pull-down
- define variables via the Environment pull-down

If you already know the trace environment that you want, skip this option and choose **Flow trace processing** or **Statement trace processing**.

---

## Procedure

Perform the following steps to display an empty user trace table.

Step	Action
1	On the Dialog Trace pop-up, select <b>Invoke trace environment with empty table</b> . <i>Result:</i> The Dialog Trace Output — Trace View panel appears.
2	Make a selection from the action bar to perform the desired preparatory action. (Action bar selections are described in “Using the Action Bar Selections” on page 50.)

# Using the Action Bar Selections

---

## Introduction

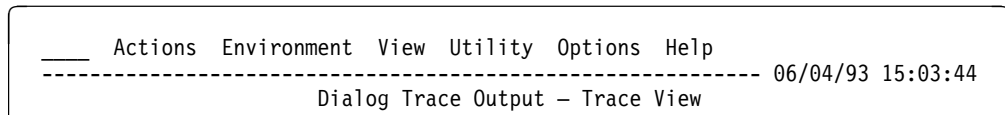
The DTF action bar allows you to modify the trace environment in a variety of ways.

Some of the action bar selections duplicate actions that you can select in other ways. For example, you can refresh a dialog using a selection from the Utility pull-down, or you can perform the same action as explained in “Refreshing a dialog (using pop-up)” on page 37.

---

## Action bar selections

The DTF action bar is displayed on the Dialog Trace Output — Trace View panel.



Each of the selections available from the action bar selections is briefly described below.

---

## Actions pull-down

The Actions pull-down menu lets you perform an action on a specific entry on the current panel. To access **Actions**, use the following procedure.

Step	Action
1	Place a slash (/) next to the table entry on which you want the selected action to be performed.
2	Press F10 to move the cursor to the action bar.
3	Type <b>A</b> and press Enter to access the Actions menu.

Step	Action
4	<p>Select one of the following options:</p> <p><b>Source member list</b> Displays the dialog source statements for those dialogs compiled with the trace option. See “Specifying the trace” on page 39 for an example of this display.</p> <p><b>Range update</b> Displays ranges defined for trace. You can then add and delete ranges from the list.</p> <p><b>Variables</b> Displays the dialog variables and their values. You can then add and delete variables from the list.</p> <p><b>Exit Dialog Trace</b> Exit the Dialog Trace Output — Trace View panel.</p>

---

### Environment pull-down

The Environment pull-down menu lets you perform an action on a dialog in the trace environment, whether or not it is listed in the table. To access **Environment**, use the following procedure.

Step	Action
1	Select <b>Environment</b> from the action bar.

Step	Action
2	<p>Select one of the following options:</p> <p><b>Flow trace</b> Turns the flow trace destinations on and off.</p> <p><b>Source member</b> Displays a list of dialogs that have been compiled for trace. You can then select the dialog whose source statements you want to view.</p> <p><b>Ranges</b> Displays a list of ranges that have been defined to dialog trace.</p> <p><b>Variables</b> Displays a list of dialog variables that have been defined to dialog trace.</p> <p><b>Trace status</b> Displays the current dialog trace status for the user.</p>

---

### View pull-down

The View pull-down menu allows you to change the view for the dialog trace table. To access **View**, use the following procedure.

Step	Action
1	Select <b>View</b> from the action bar.
2	<p>Select one of the following options:</p> <p><b>All</b> Displays both statement and flow trace entries.</p> <p><b>Flow</b> Displays only flow trace entries.</p> <p><b>Statement</b> Displays only statement trace entries.</p> <p><b>Clear trace table</b> Deletes all trace entries from the user trace table.  If you plan to rerun test dialogs after viewing trace output, you can use this option to clear the original entries from the trace table before generating new output.  <b>Note:</b> If you clear the table, the Dialog Trace pop-up appears when you next invoke DTF. (See “Selecting a Trace Type” on page 33 for instructions.)</p>

---

## Utility pull-down

The Utility pull-down menu allows you to refresh or execute a dialog. To access **Utility**, use the following procedure.

Step	Action
1	Select <b>Utility</b> from the action bar.
2	Select one of the following options:  <b>Refresh a dialog</b> Refresh a dialog.  <b>Execute a dialog</b> Execute a dialog. This may be a custom dialog written by your site to display variable names or specific debugging information that is beyond the scope of DTF.  <b>Note:</b> Do not use this option to execute a dialog for tracing. It does not produce trace output.

---

## Options pull-down

The Options pull-down menu allows you to modify the dialog trace environment. To access **Options**, use the following procedure.

Step	Action
1	Select <b>Options</b> from the action bar.
2	Select one of the following options:  <b>Preferences</b> Set user session preferences.  <b>Bypass breakpoint once</b> Bypass the remaining breakpoints for the current dialog. If you execute the same dialog later, the breakpoint dialog and destination will be invoked.  <b>Suspend trace</b> Suspend generating dialog trace output.  <b>Resume trace</b> Resume generating dialog trace output.

---

## Help pull-down

The Help pull-down menu lets you access help. To access **Help**, use the following procedure.

<b>Step</b>	<b>Action</b>
1	Select <b>Help</b> from the action bar.
2	Select one of the following options:  <b>Help for help</b> Explains how to navigate through the CT/Engine operator help panels.  <b>Help index</b> Provides an alphabetic index of help information.  <b>Glossary</b> Lists words and phrases alphabetically that have special meaning to the CT/Engine operator.  <b>About</b> Shows logo, copyright, and product version information.  <b>User information</b> Shows information about your current session, such as user ID, terminal ID, system ID, VTAM® logmode name, and current ACB name.

# Chapter 4. Putting It All Together

---

Overview ..... 56  
Dialog Trace Scenario ..... 57

## Overview

---

### Introduction

The following scenario begins with a flow trace to check the flow of dialogs and then moves into a statement trace for more detailed analysis of individual dialogs.

---

### Sample dialogs

The dialogs used in this scenario are the CL/SUPERSESSION sample dialogs KLSZPTRK and KLSZPLOG.



## Dialog Trace Scenario

---

### Introduction

You have written a new dialog, KLSZPTRK. To test it, you perform the following series of actions.

---

### Preparing to trace

You perform these steps to access the dialog trace facility.

Step	Action
1	Log onto CL/SUPERSESSION. <i>Result:</i> The CL/SUPERSESSION Main Menu appears.
2	Select <b>Features</b> on the action bar. <i>Result:</i> The Product Features pop-up appears.
3	Select <b>Dialog Trace</b> . <i>Result:</i> The Dialog Trace Enable Options pop-up appears.
4	Type <b>Yes</b> in the Keep trace environment field to ensure that the trace environment remains active if you're interrupted. You decide to accept the default of 255 trace entries on the table.
5	Press Enter. <i>Result:</i> The Dialog Trace pop-up appears.

---

### Running a flow trace

You decide to first check the dialog flow by running a trace as follows:

Step	Action
1	On the Dialog Trace pop-up, select <b>Flow trace processing</b> . <i>Result:</i> The Set Flow Destinations pop-up appears.
2	Turn on <b>Table destination</b> (for your output destination). <i>Result:</i> A pop-up message confirms your destination.
3	Press F12 twice. <i>Result:</i> You return to the application, in this case, the CL/SUPERSESSION Main Menu.

Step	Action
4	On the command line, enter <b>DIALOG KLSZPTRK</b> . <b>Result:</b> You enter the ACME Industries Personal Computer Inventory sample application.
5	Log onto the application and press Enter. <b>Result:</b> The ACME Industries Personal Computer System panel appears.
6	Press F3. <b>Result:</b> You exit the application.
7	Log onto CL/SUPERSESSION again. <b>Result:</b> The CL/SUPERSESSION Main Menu appears.
8	Reinvoke DTF on the Features pull-down by entering FD on the action bar (fast path). <b>Result:</b> The DTF dialog is reinvoked and the user trace table appears.

## Checking flow

The user trace table presents trace records of that application as well as records of whatever dialogs were concurrently executing in CL/SUPERSESSION.

```

_____ Actions Environment View Utility Options Help
----- 06/04/93 15:03:44
                Dialog Trace Output - Trace View

Enter an action or select an Action Bar item, then press Enter.
S=Source list R=Range update
                Lines 27 to 35 of 35
Dialog Mem/Copy Stmt Data                Columns 1 to 53 of 95
-----
- KLSCMD                INVOKED KLSZPTRK, SYSPARM=
- KLSZPTRK              INVOKED KLSZPLGO, SYSPARM=
- KLSZPLGO              RETURNED TO KLSZPTRK, RC=
- KLSZPTRK              INVOKED KLSZPLOG, SYSPARM=
- KLSZPLOG              RETURNED TO KLSZPTRK, RC=
- KLSZPTRK              INVOKED KLSZPINA, SYSPARM=
- KLSZPINA              INVOKED KLSZPINB, SYSPARM+
- KLSZPINB              RETURNED TO KLSZPINA, RC=
- KLSZPINA              RETURNED TO KLSZPTRK, RC=
***** BOTTOM OF DATA *****

F1=Help F2=Keys F3=Exit F7=Bkwd F8=Fwd F10=Action Bar

```

**Figure 3. Dialog Trace Output — Trace View Panel (KLSZPTRK)**

You look at the dialogs that were being executed. You decide to check dialog KLSZPLOG for errors. Before you do so, however, you turn off flow trace as follows:

Step	Action
1	On the user trace table (Dialog Trace Output - Flow View panel) action bar, select <b>Environment</b> . <i>Result:</i> The Environment pull-down appears.
2	Select <b>Flow trace</b> . <i>Result:</i> The Set Flow Destinations pop-up appears.
3	Turn the table destination off. <i>Result:</i> Flow trace ends.
4	Press F12. <i>Result:</i> You return to the user trace table.

---

### Preparing for a statement trace

You prepare for a statement trace by refreshing dialog KLSZPLOG according to the following steps.

Step	Action
1	On the user trace table (Dialog Trace Output - Flow View panel) action bar, select <b>Utility</b> . <i>Result:</i> The Utility pull-down appears.
2	Select <b>Refresh dialog</b> . <i>Result:</i> The Refresh Dialog pop-up appears.
3	Check that the Compile mode field is set to trace (default).
4	Type the dialog name, KLSZPLOG, in the Member Name field.
5	Press Enter. <i>Result:</i> A message confirms that KLSZPLOG has been refreshed.
6	Press F12 twice. <i>Result:</i> You return to the user trace table.



Step	Action
5	<p>Fill in the Add Statement Range pop-up as follows:</p> <ol style="list-style-type: none"> <li>1. Type <b>KLSZPLOG</b> as the name of the member containing the statement range to be added.</li> <li>2. Type <b>44</b> as the first statement number of the range.</li> <li>3. Type <b>32767</b> as the last statement number of the range.</li> <li>4. Type a slash in the Breakpoint destination field to specify transfer from one dialog statement to another, resulting in dynamic invocation of the breakpoint dialog.</li> </ol> <p><b>Note:</b> Output is stored in the user trace table when Breakpoint destination is selected.</p> <p>The figure below shows the fields filled in.</p> <pre> +-----+                 Add Statement Range                   Enter the following data, then press Enter.         Member . . . . . KLSZPLOG +                        Starting statement . . . . 44__                     Ending statement . . . . . 32767                    Select statement trace destination(s) with a "/"       _ Table destination                               / Breakpoint destination                           _ GTF destination                                  F1=Help  F4=Prompt  F12=Cancel                   +-----+ </pre>
6	<p>Press Enter.</p> <p><b>Result:</b> A message confirms the range that you have defined.</p>
7	<p>Press F12 twice.</p> <p><b>Result:</b> You return to the user trace table.</p>
8	<p>Press F3.</p> <p><b>Result:</b> You return to the CL/SUPERSESSSION Main Menu.</p>

## Running a statement trace

You run a statement trace with breakpoint processing for dialog KLSZPLOG by performing the following steps.

Step	Action
1	<p>On the CL/SUPERSESSION Main Menu command line, reinvoke DIALOG KLSZPTRK.</p> <p><b>Result:</b> You enter the application.</p>
2	<p>Proceed through the application.</p> <p><b>Result:</b> When dialog KLSZPLOG executes you return to the user trace table (Dialog Trace Output — Trace View panel) with the first breakpoint displayed at the bottom of the panel body. A message above the function keys confirms that a breakpoint has been encountered.</p>
3	<p>Press F3.</p> <p><b>Result:</b> The statement executes and a new breakpoint is encountered.</p>
4	<p>Continue to execute statements (via F3) until you reach statement 65.</p> <div data-bbox="573 1045 1357 1577" style="border: 1px solid black; padding: 10px;"> <pre> _____ Actions Environment View Utility Options Help ----- 06/04/93 15:0                 Dialog Trace Output - Trace View  Enter an action or select an Action Bar item, then press Enter. S=Source list R=Range update V=Variables                 Lines 172 to 184 of 184                 Columns 1 to 53 of 90 ----- Dialog  Mem/Copy  Stmt Data ----- - KLSYYYYY          SELECTED KLSBBBBB - KLSZZZZZ          INVOKED KLSBBBBB, SYSPARM= 000000000000001 - KLSBBBBB          INVOKED KLSAAAAA, SYSPARM= - KLSAAAAA          RETURNED TO KLSBBBBB, RC=1 - KLSBBBBB          INVOKED KLSCCCCC, SYSPARM= Enter,F1=Help,F12=Can - KLSCCCCC          RETURNED TO KLSBBBBB, RC= - KLSBBBBB          SELECTED KLSDDDDD - KLSZPLOG KLSZPLOG 44 set cursor userid          /* Initial c - KLSZPLOG KLSZPLOG 47 set password ''          /* Clear pas - KLSZPLOG KLSZPLOG 48 set syscsr -          /* Position - KLSZPLOG KLSZPLOG 64 set errmsg ''          /* Clear err - KLSZPLOG KLSZPLOG 65 set valid 0          /* Assume er - KLSZPLOG KLSZPLOG 66 set password (encdec('&amp;password')) /* Encrypt u Breakpoint encountered in KLSZPLOG at statement 66 F1=Help F2=Keys F3=Exit F7=Bkwd F8=Fwd F10=Action Bar </pre> </div>

---

## Checking a variable

You decide to check a variable in statement 65 of dialog KLSZPLOG, using the following procedure.

Step	Action
1	Type <b>V</b> to the left of statement 65 and press Enter. <b>Result:</b> The Trace Variables pop-up appears with a dummy entry on the table.
2	Type <b>A</b> to the left of the dummy entry and press Enter. <b>Result:</b> The Add Variable pop-up appears.
3	Fill in the fields as follows: <ul style="list-style-type: none"><li>• Type <b>KLSZPLOG</b> in the Dialog field.</li><li>• Place the cursor in the Variable field and press <b>FD</b>. <b>Result:</b> The Variable Name Selection pop-up appears.</li><li>• Select <b>VALID</b> as your variable (VALID was the variable in statement 65) and press Enter. <b>Result:</b> The Add Variable pop-up appears with the Variable field (<b>VALID</b>) and the Scope field (<b>Local</b>) filled in.</li></ul>
4	Press Enter again. <b>Result:</b> The Trace Variables pop-up appears. <b>VALID</b> is listed as the variable.
5	Press <b>F12</b> . <b>Result:</b> The user trace table appears.
6	Press <b>F3</b> . <b>Result:</b> Statement 65 executes. (A new breakpoint is encountered at statement 66.)
7	Type <b>V</b> beside statement 65 and press Enter. <b>Result:</b> You return to the Trace Variables pop-up. The value of 0 (zero) appears for the variable <b>VALID</b> .

---

## Exiting DTF

You decide that the value for variable VALID is correct. You then want to exit from DTF, using the following procedure.

<b>Step</b>	<b>Action</b>
1	Press F12. <i>Result:</i> The user trace table appears.
2	On the action bar, select <b>Options</b> . <i>Result:</i> The Options pull-down appears.
3	Select <b>Bypass breakpoint once</b> . <i>Result:</i> A message tells you that the remaining breakpoints will be bypassed once (that is, F3 will not invoke breakpoint processing when you next press it.)
3	Press F12. <i>Result:</i> You return to the user trace table.
4	Press F3. <i>Result:</i> You exit from the application.



## Chapter 5. Function and Command Reference

---

Overview .....	66
DTRACE .....	67
REFRESH .....	72
REFRESH Function .....	74

## Overview

---

### Introduction

You can use CT/Engine operator commands to perform the same functions that you can execute using the dialog trace facility. A new command and a modified command are described in this chapter.

This chapter also describes a change to the REFRESH dialog language function, which has been modified to support DTF.

---

### New command

The new DTRACE command allows you to control the dialog trace facility. This command is described on page 67.

---

### Modified command

The REFRESH command compiles a specified dialog or panel. You can now specify traceable or nontraceable mode. This command is described on page 72.

---

### Modified function

The REFRESH function now allows you to indicate whether or not the dialog to be refreshed is traceable or nontraceable. This function is described on page 74.

# DTRACE

Controls the CT/Engine dialog trace facility.

---

## Type

CT/Engine operator command.

---

## Format

**DTRACE function**  
**DEST=dest**  
**LUNAME=luname**  
**MEMBER=member**  
**RANGE=(stmtno - stmtno) | stmtno**  
**[KEEP=YES | NO]**  
**[NTD=NO | YES]**  
**[TBLSZ=255 | nnnn]**

### **function**

One of the following:

**ADD** Adds statement ranges for specified members.

**COMPEND** Ends traceable compilation mode. Once traceable compilation mode ends, any compilation generates object modules that cannot be traced.

**COMPSTRT** Starts traceable compilation mode. Once started, all compilations (for example, REFRESH) generate traceable object modules. You can override traceable compilation mode by including explicit parameters on the REFRESH command.

**DELETE** Deletes statement ranges for specified members.

**DISABLE** Disables tracing for the specified LU or nonterminal dialogs, taking either the specified or default disposition of the specified user's trace environment.

**ENABLE** Enables tracing for the specified LU or nonterminal dialogs. Once tracing is enabled, statements can be traced only after further commands or dialog functions have specified which dialogs and statements should be traced.

<b>FLOWOFF</b>	Turns off dialog flow trace for an enabled user.
<b>FLOWON</b>	Turns on dialog flow trace for an enabled user.
<b>INQ</b>	Displays global trace status, individual trace user status, DTF storage statistics, and statement ranges for specified dialogs. INQ displays information for all users when issued with no other parameters.
<b>OFF</b>	Turns off dialog tracing for all users. If KEEP=YES is specified, the users' trace environments are left intact so that tracing can resume when tracing is turned on. The contents of the interactive users' trace tables are lost when tracing is turned off.
<b>ON</b>	Turns on dialog tracing. Tracing must be turned on before functions controlling tracing are accepted, except for COMPSTRT, COMPEND, and INQ.
<b>LUNAME</b>	Terminal LU name.  After trace is enabled for a specified LU name, the same LU name must be used to identify the user in DTRACE commands issued to control tracing for that user. This parameter is required with NTD=NO.
<b>NTD</b>	Specifies whether the command applies to nonterminal dialogs or an LU name.
<b>YES</b>	The command applies to nonterminal dialogs. NTD=YES and LUNAME parameters are mutually exclusive. DEST=BP and DEST=TABLE are invalid when NTD=YES.
<b>NO</b>	The command applies to an LU name. When NTD=NO, the LUNAME parameter is required.
<b>DEST</b>	The destination for trace output. Valid values are:
<b>BP</b>	Execution of the statement causes dynamic invocation of the breakpoint dialog. Output to the table destination is implicit with BP.
<b>TABLE</b>	Trace output is written to the interactive user's trace table.

**GTF** Trace output is written to GTF. To write data to GTF, you must also start GTF and use CT/Engine operator commands GTF and GTRACE.

**KEEP** Determine whether or not the user's trace environment is retained at logoff or when tracing is disabled.

**YES** The user's trace environment is retained.

**NO** The user's trace environment is deleted.

**Defaults:** If ENABLE is specified, the default is KEEP=YES. If DISABLE is specified, the default is the value of KEEP when ENABLE was last specified.

**MEMBER** Source member name.

**RANGE** Starting and ending statement numbers within the source dialog. Statement numbers must be enclosed by parentheses and separated by a dash if starting and ending numbers are specified. If only a starting statement number is specified, you can omit the parentheses and dash.

With the ADD parameter, if the starting statement number is specified and the ending statement number is omitted, the range consists of one statement.

With the DELETE parameter, if the ending statement number is omitted, all ranges declared with the same starting statement number are deleted for the specified destination. If RANGE is omitted, all ranges for the given dialog are deleted. If DEST is omitted, applicable ranges for all destinations are deleted.

**TBLSZ** Indicates the maximum number of trace entries in the user trace table.

**Note:** A single entry has a maximum size equal to the source statement length plus 20 bytes. This results in an average size of approximately 50 bytes per entry.

---

## Usage Notes

1. The allowable forms of DTRACE are:

```
DTRACE ADD LUNAME=luname | NTD=YES
          MEMBER=member
          RANGE=(nn-nn) | nn
          DEST=dest
```

```
DTRACE COMPEND
```

```
DTRACE COMPSTRT
```

```
DTRACE DELETE LUNAME=luname | NTD=YES
              MEMBER=member
              RANGE=(nn-nn) | nn
              DEST=dest
```

```
DTRACE DISABLE LUNAME=luname | NTD=YES
               KEEP=YES | NO
```

```
DTRACE FLOWON LUNAME=luname | NTD=YES
              DEST=dest
```

```
DTRACE FLOWOFF LUNAME=luname | NTD=YES
```

```
DTRACE INQ LUNAME=luname | NTD=YES
```

```
DTRACE OFF
```

```
DTRACE ON
```

2. The COMPSTRT, COMPEND, INQ, and ON functions are the only functions that can be used while tracing is turned off.
3. Statement numbers for a range need not correspond exactly to SSPL procedural statements. For example, statement numbers may represent comment lines, or the ending statement number may be greater than the last statement number of a source member.

---

## Example

To display trace status for the system and the status and ranges for each user, issue:

```
DTRACE INQ
```

*CT/Engine Messages* has the *KLVO Pnnn* messages that are issued by the DTRACE command, and a description of the data contained in them.

---

**See Also**

“REFRESH” on page 72.

## REFRESH

Loads a new copy of a dialog, module, or sense code table, and makes it available for subsequent execution requests.

---

### Type

CT/Engine operator command.

---

### Format

**REFRESH** {**PANEL** | **DIALOG**} **name**  
[**TRACE** | **NOTRACE**]

**MODULE** **name**

**SENSE** **name** **dsname**

**PANEL** Specifies that a dialog in the TLVPNLS concatenation is to be refreshed. *DIALOG* may be used interchangeably with *PANEL*. *P* and *D* may be used as abbreviations.

**MODULE** Specifies that a module in the TLVLOAD concatenation is to be refreshed. *M* may be used as an abbreviation.

**SENSE** Specifies that the global sense table is to be refreshed. *S* may be used as an abbreviation.

**name** The 1–8 character name of the dialog, module, or sense code member that is to be refreshed.

**dsname** The 1–44 character data set name that contains the sense code rule definition member, *name*.

**TRACE** Specifies that the dialog, *name*, is to be compiled with additional information that may be used by dialog trace.

**NOTRACE** Specifies that no dialog trace information is to be included in the compilation of *name*.

---

### Usage Notes

1. If neither *TRACE* nor *NOTRACE* is specified, the default is based on the TRCCOMP keyword in the Dialog Manager initialization member, KLVINDM, in TLV Parm



2. If you update a dialog in your panel library, REFRESH causes the updated version to be used for subsequent invocations. If the modified dialog fails to compile during REFRESH, the old dialog is still available for execution.
3. *dsname* must be enclosed in single quotation marks if the REFRESH command is issued from the MVS console.
4. REFRESH MODULE is effective only for programs invoked by the LINK dialog function. It will not replace user exits or CT/Engine programs.

---

**Example**

To refresh panel LOGNT04 in your panel library, enter:

```
REFRESH P LOGNT04
```

Dialog trace information will be included if KLVINDM requested it. To ensure no extra data is included:

```
REFRESH PANEL XYZ NOTRACE
```

To refresh the global sense table located in MYLIB.USER.SENSETBL(SNSIN):

```
REFRESH S SNSIN MYLIB.USER.SENSETBL
```

---

**See Also**

“DTRACE” on page 67.

# REFRESH Function

---

## Description

Loads and compiles a new copy of the specified dialog name.

---

## Type

Dialog language function.

---

## Format

**REFRESH(dialog  
[bool])**

**dialog**      Name of the dialog to be reloaded.

**bool**          A boolean expression indicating traceability:

**0 (FALSE)**    The specified dialog is not traceable.

**Not 0 (TRUE)** The specified dialog is traceable.

---

## Return Codes

**0 (FALSE)**    The specified dialog did not compile successfully.

**1 (TRUE)**     The specified dialog was successfully refreshed.

---

## Usage Notes

1. Compilation errors appear in TLVLOG
2. If *bool* is omitted, the compilation is performed according to the current traceable compilation mode.



## CL/ENGINE Messages

---

### Introduction

CT/Engine messages for the dialog trace facility follow in alphanumerical order.

---

### Messages

- KLVD001 LUNAME, *luname*, NOT UNIQUE AND WILL NOT TRACE**  
**Explanation:** A duplicate LUNAME exists on the system, and tracing of *luname* will not occur.  
**System Action:** *luname* will not be traced.  
**User Response:** Rename *luname* so that it is unique to the system.  
**Message Type:** INFO.
- KLVD002 DIALOG TRACE ENVIRONMENT INTEGRITY ERROR**  
**Explanation:** There is a severe problem with the DTF environment.  
**System Action:** None.  
**User Response:** Contact Candle Support Services.  
**Message Type:** ERROR.
- KLVP411 DIALOG TRACE STATUS(ON|OFF) TRACEABLE COMPILATION MODE(ON|OFF)**  
**Explanation:** Dialog trace is on so that tracing of dialogs will occur, or it is off so that tracing of dialogs will not occur.  
Traceable compilation mode is on so that compilation creates an object module that is traceable, or it is off so that compilation creates an object module that is not traceable.  
**System Action:** None.  
**User Response:** None.  
**Message Type:** REPLY.
- KLVP412 DIALOG TRACE STORAGE ALLOCATED *nnnn* KB**  
**Explanation:** The amount of storage used by DTF for the retention of source statements and control blocks is displayed.  
**System Action:** None.  
**User Response:** None.  
**Message Type:** REPLY.

- KLVOP413** *luname*|NTD ENABLED|DISABLED  
[TBLSZ(*nnnn*)]SUSPENDED  
**Explanation:** The specified LU or a nonterminal dialog (NTD) was enabled or disabled. If interactive tracing capabilities were granted, user trace table size, *nnnn*, is displayed. For interactive users, suspension of output to the table destination may be indicated with SUSPENDED.  
**System Action:** None.  
**User Response:** None.  
**Message Type:** REPLY.
- KLVOP414** *luname*|NTD MEMBER(*memname*) RANGE(*start-end*)  
DEST(BP|GTF|TABLE)  
**Explanation:** A range of statements, *start* through *end*, will be traced from source member *memname* for dialogs (which copy the member and have been compiled for trace) executing at either the specified LU or as nonterminal dialogs. The dialog manager generates trace output to BP, GTF, or TABLE.  
**System Action:** None.  
**User Response:** None.  
**Message Type:** REPLY.
- KLVOP415** *luname*|NTD FLOWON DEST(BP|GTF|TABLE)  
**Explanation:** FLOW tracing is active for the specified LU or for a nonterminal dialog (NTD). The dialog manager generates trace output to BP, GTF, or TABLE.  
**System Action:** None.  
**User Response:** None.  
**Message Type:** REPLY.
- KLVOP416** DIALOG TRACE IS OFF|ON  
**Explanation:** Indicates the status of the dialog trace facility.  
**System Action:** If trace was turned on, the DTRACE command can be used to enable terminals or nonterminal dialogs and to add and delete ranges. Commands specifying users, dialogs, and ranges to be traced are rejected if tracing was turned off. When trace is turned on, all enabled LUs and nonterminal dialog resources can generate trace output.  
**User Response:** None.  
**Message Type:** INFO.
- KLVOP417** DESTINATIONS BP AND TABLE INVALID FOR  
NON-TERMINAL DIALOGS AND NON-INTERACTIVE  
USERS  
**Explanation:** A command specifying the BP or TABLE destination was issued for a noninteractive user or for nonterminal dialogs. These output destinations are valid only for interactive users enabled by LU name.

**System Action:** The command completes unsuccessfully.  
**User Response:** Reissue the command using a different destination or enable the terminal for interactive tracing.  
**Message Type:** INFO.

**KLVOP418** *luname*|NTD ENABLED FOR DIALOG TRACE  
TBLSZ(*nnnn*)

**Explanation:** The specified LU or a nonterminal dialog was enabled for tracing of dialogs. When a terminal identified by *luname* is granted interactive capabilities, user trace table size, *nnnn*, is displayed.

**System Action:** Commands affecting the enabled LU or nonterminal dialogs are accepted.

**User Response:** None.

**Message Type:** INFO.

**KLVOP419** *luname*|NTD DISABLED FOR DIALOG TRACE,  
ENVIRONMENT RETAINED|DELETED

**Explanation:** Tracing for the specified LU or the nonterminal dialogs was disabled. The trace environment is deleted or retained.

**System Action:** When the trace environment is deleted, all dialogs and ranges that were declared traceable for the user are no longer traceable, even after the user is re-enabled.

**User Response:** None.

**Message Type:** INFO.

**KLVOP420** RANGE(*start-end*) ADDED FOR *luname*|NTD:  
MEMBER(*member*) DEST(BP|GTF|TABLE)

**Explanation:** A range of statements, *start* through *end*, was added for the specified LU, specified member and specified destination, or for the nonterminal dialogs (NTD).

**System Action:** When statements within the range for *member* are executed, trace output can be directed to a trace output destination: BP, GTF, or TABLE.

**User Response:** None.

**Message Type:** REPLY.

**KLVOP421** RANGE(*start-end*|ALL) DELETED FOR *luname*|NTD  
MEMBER(*member*|ALL)

**Explanation:** A range of statements, *start* through *end*, has been deleted for the specified LU or nonterminal dialogs (NTD), specified or all members, and specified or all destinations.

**System Action:** The specified range that was deleted for the specified member, destination, and LU or NTD will no longer be traced.

**User Response:** None.

**Message Type:** REPLY.

- KLVOP422 TRACEABLE COMPILATION MODE IS ON|OFF**  
**Explanation:** Indicates the status of traceable compilation mode.  
**System Action:** Output from automatic compilation and the REFRESH command using parameter defaults is a traceable dialog if the mode is ON; it is nontraceable if the mode is OFF.  
**User Response:** None.  
**Message Type:** INFO.
- KLVOP423 COMMAND INVALID WHILE DIALOG TRACE IS OFF**  
**Explanation:** A command was issued that requires that DTF be activated.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Turn on DTF and reissue the command.  
**Message Type:** ERROR.
- KLVOP424 *luname*|NTD NOT ENABLED FOR TRACING**  
**Explanation:** An attempt was made to disable an LU or nonterminal dialogs (NTD), but the command completed unsuccessfully because the LU or NTD was not enabled for tracing.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Correct the LU name and reissue the command.  
**Message Type:** ERROR.
- KLVOP425 RANGE(*start-end*|ALL) DOES NOT EXIST FOR *luname*|NTD MEMBER(*member*|ALL) DEST(BP|GTF|TABLE|ALL)**  
**Explanation:** The statement ranges being deleted for the specified LU or nonterminal dialogs (NTD), specified members, and specified destinations do not exist.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Use the DTRACE command to display the correct range information and reissue the command.  
**Message Type:** ERROR.
- KLVOP426 TRACEABLE COMPILATION MODE UNCHANGED**  
**Explanation:** A command was issued to change the compilation mode to the mode that already exists.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Reissue the command to set the desired mode. If the desired mode is the current mode, no user response is necessary.  
**Message Type:** ERROR.

- KLVOP427 FLOW TRACE STATUS UNCHANGED**  
**Explanation:** A command was issued to change the state of the flow trace to the state that already exists.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Reissue the command to set the flow trace to the desired state, or do nothing if the desired state is the current one.  
**Message Type:** ERROR.
- KLVOP428** *keyword = value* **INVALID**  
**Explanation:** A command containing an invalid parameter keyword was issued.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Consult the command documentation, and then reissue the command.  
**Message Type:** ERROR.
- KLVOP430** *luname|NTD* **ALREADY ENABLED FOR TRACING**  
**Explanation:** An attempt was made to enable an LU or nonterminal dialog (NTD), but the LU or NTD was already enabled for tracing.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Correct the LU name or disable the LU or nonterminal dialogs, and reissue the command.  
**Message Type:** ERROR.
- KLVOP431 STORAGE ISOLATION POOL NOT INITIALIZED**  
**Explanation:** A storage isolation pool could not be initialized during DTF initialization.  
**System Action:** DTF is not on.  
**User Response:** Use the STORAGE command to display storage utilization. Adjust storage allocation parameters and recycle CT/Engine if storage was underallocated. Otherwise, contact Candle Support Services.  
**Message Type:** ERROR.
- KLVOP432 DIALOG TRACE HANDLE POOL NOT INITIALIZED**  
**Explanation:** An attempt to turn DTF on has been made, but a handle pool could not be initialized.  
**System Action:** DTF is not turned on.  
**User Response:** Contact Candle Support Services.  
**Message Type:** ERROR.



- KLVOP433 TRACE ENVIRONMENT INTEGRITY ERROR**  
**Explanation:** Data structures used for representation of dialog trace information are corrupted.  
**System Action:** DTF is turned off and cannot be restarted.  
**User Response:** Contact Candle Support Services.  
**Message Type:** ERROR.
- KLVOP434 LUNAME= AND NTD= PARAMETER CONFLICT**  
**Explanation:** The LUNAME and NTD keyword parameters are mutually exclusive.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Reissue the command with either the LUNAME or NTD keyword parameter.  
**Message Type:** ERROR.
- KLVOP441 INTERACTIVE DIALOG TRACING INVALID FOR NTD**  
**Explanation:** An attempt to enable nonterminal dialogs for tracing was made with the TBLSZ parameter specified.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Reissue the command, omitting the TBLSZ keyword parameter.  
**Message Type:** ERROR.
- KLVOP443 *luname*|NTD FLOWOFF DEST(BP|GTF|TBL)**  
**Explanation:** Flow trace has been turned off for the specified LU or nonterminal dialogs (NTD) and for the specified destination.  
**System Action:** Transfers of control between dialogs for NTD or *luname* will no longer be traced.  
**User Response:** None.  
**Message Type:** INFO.
- KLVOP444 DIALOG TRACE IS ALREADY ON**  
**Explanation:** An attempt was made to activate DTF, but it was already activated.  
**System Action:** The command completes unsuccessfully.  
**User Response:** None.  
**Message Type:** ERROR.
- KLVOP445 UNEXPECTED RETURN CODE *nn* FROM KLV\$TEM**  
**Explanation:** KLV\$TEM1 returned an unanticipated return code.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Contact Candle Support Services.  
**Message Type:** ERROR.

- KLVOP446** **TRACING ALREADY SUSPENDED|RESUMED**  
**Explanation:** An attempt was made to suspend or resume dialog trace, but it was already suspended or resumed.  
**System Action:** The command completes unsuccessfully.  
**User Response:** None.  
**Message Type:** ERROR.
- KLVOP447** **RANGE (start-end) FOR luname|NTD MEMBER member, DEST dest, ALREADY EXISTS**  
**Explanation:** An attempt was made to add a specified range that already exists.  
**System Action:** The command completes unsuccessfully.  
**User Response:** None.  
**Message Type:** ERROR.
- KLVOP448** **LU luname NOT ENABLED FOR INTERACTIVE TRACING**  
**Explanation:** An attempt was made to suspend tracing for *luname* or an NTD that was not enabled for interactive tracing.  
**System Action:** The command completes unsuccessfully.  
**User Response:** None.  
**Message Type:** ERROR.
- KLVOP450** **nnnnnn, A REQUIRED PARAMETER, IS MISSING**  
**Explanation:** A command was issued without a required parameter.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Reissue the command with required parameter.  
**Message Type:** ERROR.
- KLVOP451** **LUNAME= OR NTD=YES ARE REQUIRED**  
**Explanation:** A command was issued that requires the LUNAME or NTD keyword parameter.  
**System Action:** The command completes unsuccessfully.  
**User Response:** Reissue the command for a specified LU or NTD.  
**Message Type:** ERROR.
- KLVOP452** **TBLSZ= IGNORED FOR REENABLE**  
**Explanation:** A nonzero trace table size was specified for a disabled, interactive trace user with a retained trace environment. The trace table size cannot be changed when the user is re-enabled.  
**System Action:** The command completes successfully, but the newly-specified trace table size is ignored.  
**User Response:** If a new trace table size is desired, disable the user with the KEEP=NO option to delete the user's trace

environment. Then re-enable the user with a new trace table size.

**Message Type:** INFO.

**KL VOP703 DIALOG *dlgname* REFRESHED (TRACE|NOTRACE)**

**Explanation:** Dialog *dlgname* specified in the REFRESH command was successfully refreshed. TRACE is indicated if the current compilation mode is traceable, or an explicit traceable REFRESH overrides a nontraceable compilation mode.

**System Action:** None

**User Response:** None.

**Message Type:** REPLY.

**KL VDM055 COMPILATION MODE IS TRACEABLE|NON-TRACEABLE**

**Explanation:** Indicates the status of compilation mode set during dialog manager initialization.

**System Action:** Output is a traceable or nontraceable dialog, depending on the compilation mode.

**User Response:** None.

**Message Type:** INFO.

**KL VDM134 STATEMENT RESOURCE INITIALIZATION FAILED**

**Explanation:** The dialog manager was unable to establish the traceability of statement ranges for a dialog user who was enabled for tracing.

**System Action:** Dialogs execute, but no tracing occurs.

**User Response:** Contact Candle Support Services.

**Message Type:** ERROR.



## Appendix A. Generalized Trace Facility Output

---

### Formatted Output

---

#### Dialog trace output

When GTF has been started and the command GTF ON is issued, dialog trace output can be directed to GTF through any of the following means:

- the DTRACE basic operator command
- the CUA operator menus
- the DTRCNTL SSPL function
- the KLVDTRC dialog

Instructions for GTRACE and the formatting of GTF output are in the *Operator's Guide* and the *Problem Determination Guide*.

---

#### GTF trace output

An example of GTF trace output from the dialog trace facility follows.

```

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0001 SG0001 LU M722BA4
LRN 001C42FE  DLG GATEWAY FLOW
INVOKED XYZNIEN , SYSPARM= TDLC81

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0002 SG0001 NTD
LRN 001C42FE  DLG XYZNIEN MEM XYZNIE SNUM 00114
if &rc = 4

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0003 SG0001 LU M722BA4
LRN 001C42FE  DLG XYZNIEN MEM XYZNIE SNUM 00132
IF &natdflt!=&ABCuser

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0004 SG0001 LU M722BA4
LRN 001C42FE  DLG XYZNIEN MEM XYZNIE SNUM 00133
do

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0005 SG0001 LU M722BA4
LRN 001C42FE  DLG XYZNIEN MEM XYZNIE SNUM 00134
set btzgprof '&natdflt..GROUP.COMMON.PROFILE'

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0006 SG0001 LU M722BA4
LRN 001C42FE  DLG XYZNIEN FLOW
RETURNED TO GATEWAY , RC= 12

```

---

### GTF trace output from an encrypted source

When the corresponding source statement is encrypted, the output for a source statement is replaced with the constant:

**\*\*\* ENCRYPTED SOURCE STATEMENT \*\*\***

The following is an example of GTF trace output from the dialog trace facility when the statements from the source dialog are encrypted.

```

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0001 SG0001 LU M722BA4
LRN 001C42FE DLG GATEWAY FLOW
INVOKED XYZNIEN , SYSPARM  TDLC81

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0002 SG0001 NTD
LRN 001C42FE DLG XYZNIEN MEM XYZNIE SNUM 00114
*** ENCRYPTED SOURCE STATEMENT ***

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0003 SG0001 LU M722BA4
LRN 001C42FE DLG XYZNIEN MEM XYZNIE SNUM 00132
*** ENCRYPTED SOURCE STATEMENT ***

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0004 SG0001 LU M722BA4
LRN 001C42FE DLG XYZNIEN MEM XYZNIE SNUM 00133
*** ENCRYPTED SOURCE STATEMENT ***

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0005 SG0001 LU M722BA4
LRN 001C42FE DLG XYZNIEN MEM XYZNIE SNUM 00134
*** ENCRYPTED SOURCE STATEMENT ***

USR4C 3E8 ASCB 02149370  JOBK KLV
SQ0006 SG0001 LU M722BA4
LRN 001C42FE DLG XYZNIEN FLOW
RETURNED TO GATEWAY, RC= 12

```

## Output keywords

The following keywords are used in GTF output.

<b>Table 4 (Page 1 of 2). GTF Output Keywords</b>	
<b>Keyword</b>	<b>Definition</b>
ASCB	The address space control block address for the CT/Engine address space.
JOBK	The job name of the CT/Engine address space.
SQ	The sequence number of the trace entry.
SG	The segment number for a trace entry that requires multiple segments when there is more data than can be displayed in a single segment.
LRN	The logical resource number of the dialog thread that caused the trace output.
LU	The LU name of the terminal at which the traced dialog executed.
NTD	The trace data was output for a nonterminal dialog.

<b>Table 4 (Page 2 of 2). GTF Output Keywords</b>	
<b>Keyword</b>	<b>Definition</b>
DLG	The name of the dialog that was executing.
MEM	The name of the member that was executing.
SNUM	The statement number of the statement that was executing.
FLOW	Indicates that the last line of data is flow trace data. <b>Note:</b> If FLOW is not present, the last line of data is statement trace data.

---

## **SYSPARM**

Whenever SYSPARM is a packed variable, the following constant is displayed in place of the value of SYSPARM:

**\*\*\* PACKED STRUCTURE \*\*\***

In flow trace entries the SYSPARM= and RC= values are omitted when their representation is a string of length 0. Numeric values are represented as strings of 11 characters, of which the first character is either zero (0) or a minus/hyphen (-).



## A

Add Statement Range pop-up 39, 61  
adding statement ranges 41  
adding variables 41, 48

## B

breakpoint 19, 61  
    bypassing 46, 53  
    deactivating 45  
    destination 34, 35, 40, 42, 44  
    processing 20, 44

## C

Candle Electronic Customer Support (CECS) 10  
CECS  
    *See* Candle Electronic Customer Support (CECS)  
clearing the trace table 52  
commands, CL/ENGINE  
    STORAGE DETAIL 20  
compilation mode  
    *See* traceable compilation mode  
compiled dialog storage 20  
compiling a new dialog copy 74  
conventions, documentation 12  
customer support 10

## D

deactivating breakpoint 45  
deleting entries from trace table 52  
deleting ranges 46  
deleting variables 48  
dialog  
    executing 53  
    KLVDTTC 28, 35, 42, 44  
    loading and compiling a new copy 74  
    refreshing 37, 53, 74

dialog trace facility, defined 18  
Dialog Trace Output — Trace View panel 50  
    *See also* user trace table  
Dialog Trace pop-up 33  
displaying  
    dialogs compiled for trace 52  
    executable dialog statements 39  
    flow entries 52  
    statement entries 52  
    statement ranges 41, 51, 52  
    trace status 52  
    variables 41, 51, 52  
documentation conventions 12  
documentation set 14  
DTF dialog  
    *See* dialog, KLVDTTC  
DTRACE 67  
DTRACE Commands pop-up 25, 26

## E

electronic customer support 10  
empty-table option 49  
enabling your terminal 31  
ending  
    *See* turning off  
entries, deleting 52  
error messages 75–83  
establishing a trace environment 30  
examining flow trace data 35  
examining statement trace data 41  
executing a dialog 53  
exiting Trace View panel 51

## F

features, DTF 18  
flow trace 19  
    examining data 35  
    performing 34  
    setting destination 40  
    turning on/off 46, 52

function 74

## G

generalized trace facility (GTF)  
output 20, 85–88  
    encrypted 86  
    keywords 87  
    SYSPARM 88

## H

help, described 54

## I

implementing DTF 18  
interactive tracing  
    *See* breakpoint  
invoking trace environment with empty table 49

## K

KLVDTRC dialog 28

## L

loading a new dialog copy 74

## M

maintaining variables 48  
messages  
    CL/ENGINE 76–83  
modifying trace environment 50  
modifying variables 48

## O

operator commands 66–74  
output  
    destination 34, 40  
    suspending and resuming 53  
    types 20

## P

panels  
    Add Statement Range 39, 61  
    Dialog Trace 33  
    Dialog Trace Output — Trace View 35, 42,  
        50, 58, 62  
    REFRESH Dialog 37  
    Set Flow Destinations 34  
    Source Statement List 39, 60  
    Trace Variables 48  
performing a trace  
    establishing a trace environment 30  
    flow trace 34  
    statement trace 37

## R

range storage 21  
    trace procedure  
ranges, deleting 46  
REFRESH 72  
REFRESH Dialog pop-up 37  
REFRESH function 74  
refreshing a dialog 37, 53  
    creating a CLIST 38  
resuming trace output 53

## S

security 25, 27, 28  
selecting the dialog 39  
Set Flow Destinations pop-up 34  
setting user preferences 53  
Source Statement List pop-up 39, 60  
    panel actions 41  
starting  
    *See* ?  
    *See* turning on

- statement ranges 41, 51, 52
- statement trace 19
  - breakpoint processing 19
  - examining data 41, 61
  - performing 37
  - specifying the trace 39
- stopping
  - See* stopping a trace
  - See* turning off
- stopping a trace
  - bypassing breakpoint 46
  - deleting ranges 46
  - suspending trace output 45
  - turning off flow trace 46
- storage
  - options 20
  - requirements 20
- STORAGE DETAIL command 20
- support
  - See* customer support
- suspending trace output 45, 53
- symbols, use of 13
- SYSPARM 35, 88

## T

- table
  - See* user trace table
- table storage 21
- telephone support 10
- trace entries, deleting 52
- trace environment
  - establishing 30
  - invoking with empty table 49
  - modifying 50
  - retaining 31
- trace output 20
  - See also* output
- trace table
  - See* user trace table
- trace types 33
  - flow trace 19
  - statement trace 19
- Trace Variables pop-up 48
  - panel actions 48
- Trace View action bar
  - Actions pull-down 50
  - Environment pull-down 51
  - Help pull-down 54

- Trace View action bar (*continued*)
  - Options pull-down 53
  - Utility pull-down 53
  - View pull-down 52
- traceable compilation mode 37
  - ending 26
  - starting 26
- transfer of control 19
- turning off
  - See also* stopping a trace
  - DTF 25
  - traceable compilation mode 26
- turning on
  - DTF 25
  - traceable compilation mode 26

## U

- updating a dialog 74
- user session preferences 53
- user trace table
  - action bar 50
  - empty 49
  - entries 31
  - full 45
  - output 20
  - panel actions
    - range update 36, 43
    - source list 36, 43
  - preventing overwrite 45
  - storage 20

## V

- variables
  - adding 41, 48
  - deleting 48
  - displaying 41
  - maintaining 48
  - modifying 48
- viewing trace data
  - flow trace 35
  - statement trace 41

