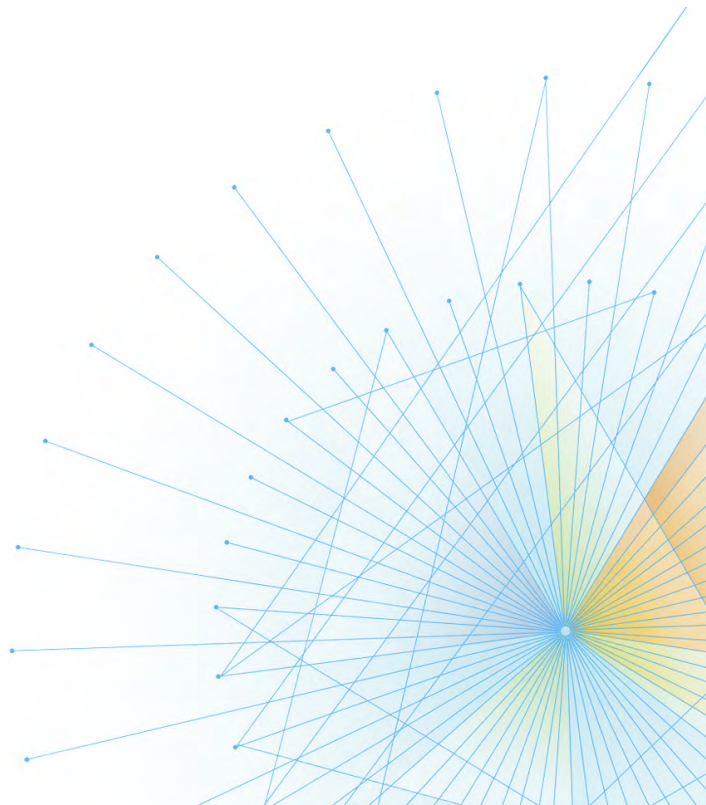




The Mainframe Software Partner For The Next 50 Years

# Xpediter/TSO and Xpediter/IMS Quick Reference

**Release 17.02**



Please direct questions about Xpediter/TSO and Xpediter/IMS  
or comments on this document to:

**Compuware Support Center**

**<https://go.compuware.com/>**

This document and the product referenced in it are subject to the following legends:

Copyright 1996-2019 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS-Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

Xpediter, Code Coverage, File-AID, and Abend-AID are trademarks or registered trademarks of Compuware Corporation.

CICS, DB2, IBM MQ for z/OS, and IBM are trademarks of International Business Machines Corporation.

Adobe® Reader® is a trademark of Adobe Systems Incorporated in the United States and/or other countries.

All other company and product names are trademarks or registered trademarks of their respective owners.

# Contents

---

<b>Introduction</b> . . . . .	<b>.vii</b>
Intended Audience . . . . .	vii
Related Publications . . . . .	vii
Notation Rules . . . . .	viii
Reading the Syntax Diagrams . . . . .	viii
<b>Common Parameters</b> . . . . .	<b>1</b>
location . . . . .	1
data . . . . .	1
CSR . . . . .	1
keyword . . . . .	1
literal . . . . .	2
conditional-expression . . . . .	2
<b>PF Key Settings</b> . . . . .	<b>2</b>
<b>Primary Commands</b> . . . . .	<b>3</b>
AA SNAP . . . . .	3
ACCEPT . . . . .	3
AFTER . . . . .	3
ALLOCATE . . . . .	4
AT . . . . .	4
BEFORE . . . . .	4
BOTTOM . . . . .	5
BROWSE . . . . .	5
CCHILITE . . . . .	5
CONNECT . . . . .	5
COUNT . . . . .	5
COVER . . . . .	6
DELETE . . . . .	7
DISC . . . . .	8
DLEFT . . . . .	8
DLI . . . . .	8
DOWN . . . . .	9
DRIGHT . . . . .	9
DROP . . . . .	9
END . . . . .	9
EXCLUDE . . . . .	10
EXIT . . . . .	10
FADB2 . . . . .	11
FIND . . . . .	11
GEN . . . . .	13
GETMAIN . . . . .	14
GO . . . . .	14
GOBACK . . . . .	15
GOTO . . . . .	15
GPREGS . . . . .	15
HELP . . . . .	16

IF	16
INCLUDE	16
INSERT	16
INTERCEPT	17
KEEP	17
LEFT	18
LINE	18
LOAD	19
LOCATE	19
LOG	19
MEMORY	19
MONITOR	20
MOVE	20
NOLINES	20
ONETIME	21
PAUSE	21
PEEK	21
PSEUDOSOURCE	22
RESET	22
RESUME	22
RETEST	22
RETURN	23
REVERSE	23
RIGHT	23
RUN	23
RUNTO	24
SCRNSAVE	24
SET	24
SHOW	29
SKIP	30
SOURCE	30
STATUS	31
TEST	31
TOGGLE	31
TOP	32
TRACE	32
UP	32
USE	33
USING	33
VERIFY	33
WHEN	33
WHEREIS	34
WS	34
XCHANGE	34
XPED	35
<b>Line Commands</b>	<b>35</b>
) , )n , )) , ))n	35
( , (n , (( , ((n	35
:	35
:n	35

A, AA	35
B, BB	36
C, CC	36
D	36
D:	36
DA	36
DB	36
DC	36
DD	37
DE	37
DG	37
DH	37
DO	37
DS	37
DT	37
DV	37
E, En, EE	38
F, Fn	38
G, GG	38
GT	38
H, Hn, HH	38
I, In	38
K, Kn, KK	38
KE, KEn	39
KH, KHn	39
K*	39
L, Ln	39
M	39
O, OO	39
P, Pn, PP	39
PE, PEn	39
PH, PHn	39
P*	40
S, SS	40
T, TT	40
V, VV	40
X, XX	40
XP	40
Z	40
<b>Link-Edit Options</b>	<b>41</b>
<b>Unattended Batch Commands</b>	<b>41</b>



---

# Introduction

This document summarizes the PF key settings, commands, and link-edit options for Xpediter/TSO, Xpediter/IMS, and Xpediter for DB2 Extension.

---

## Intended Audience

This Quick Reference is written for application programmers engaged in program development and maintenance. A working knowledge of COBOL, Assembler, PL/I, or C, and your local system environment is assumed.

---

## Related Publications

The following Compuware documents are also available for use with the Xpediter/TSO and Xpediter/IMS product:

- *Xpediter/TSO and Xpediter/IMS Installation and Configuration Guide*
- *Xpediter/TSO and Xpediter/IMS Advanced Configuration Guide*
- *Xpediter/TSO and Xpediter/IMS COBOL User Guide*
- *Xpediter/TSO and Xpediter/IMS Assembler User Guide*
- *Xpediter/TSO and Xpediter/IMS PL/I User Guide*
- *Xpediter/TSO and Xpediter/IMS C Language User Guide*
- *Xpediter/TSO and Xpediter/IMS Reference Manual*
- *Xpediter/TSO and Xpediter/IMS Messages and Codes*.

## Online Documentation

The Xpediter/TSO and Xpediter/IMS product installation package does not include the product documentation. Access the Xpediter/TSO and Xpediter/IMS documentation from the Compuware Support Center website at <https://go.compuware.com> in the following electronic formats:

- Release Notes in HTML format
- Product manuals in PDF format
- Product manuals in HTML format.

The product documentation is available for viewing or downloading:

- View PDF files with the free Adobe Reader, available at <http://www.adobe.com>.
- View HTML files with any standard web browser.

---

## Notation Rules

The following notation rules are used throughout this document:

- Command names are shown in syntax diagrams with both the full name and all possible abbreviations.
- A parameter is either a keyword or a variable.
  - Maximum strings for keywords are shown in UPPERCASE characters and must be spelled exactly as shown. The remaining characters of the keyword can be included at your discretion.
  - Variables are user-specified values and are printed in *lowercase italics*. For example, *dsname* indicates you are to substitute a value.

---

## Reading the Syntax Diagrams

Syntax diagrams define primary command syntax.

A **parameter** is either a keyword or a variable.

All KEYWORDS are shown in uppercase characters and must be spelled exactly as shown. You cannot substitute another value. If any part of a KEYWORD is shown in lowercase characters, that part is optional.

Variables are user-specified values and are printed in lowercase italics. For example, *dataset-name* indicates you are to substitute a value.

The syntax for commands is described in diagrams that help you visualize parameter use. The following example shows a command and a parameter:

▶▶—COMMAND—parameter—▶▶

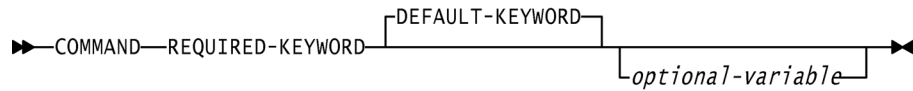
Read the diagrams from left to right and from top to bottom. These symbols help you follow the path of the syntax:

- ▶▶ indicates the beginning of a statement.
- indicates the statement is continued on the next line.
- ▶ indicates the statement is continued from the previous line.
- ▶ indicates the end of a statement.

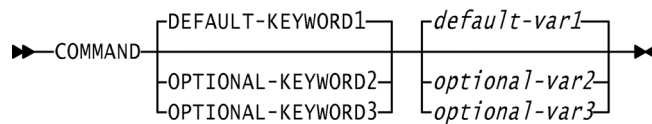
Required parameters appear on the horizontal line (the main path). Optional parameters appear below the main path. Default parameters appear above the



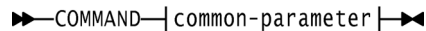
main path and are optional. The command will execute the same whether the default parameter is included or not.



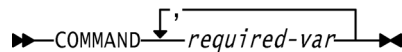
Vertically stacked parameters are mutually exclusive. If you must choose a parameter, one item of the stack appears on the main path. If the parameters are optional, the entire stack appears below the main path. If a parameter in a stack is the default, it appears above the main path.



If the same parameters are used with several commands, their syntax may be documented in a separate diagram. In the command syntax, these common parameters are indicated with separators before and after the parameter name.



An arrow returning to the left indicates a repeatable item. If the arrow contains a comma, separate the repeated items with a comma.





---

# Common Parameters

The following common parameters are used throughout this document.

---

## location

A place at which breakpoints are set or deleted. The following types of locations are valid:

statement-number	ALL [STATE]
paragraph-name	ALL PARA
program-name	ALL PROG
module-name	ALL MOD
procedure-name	ALL PROC
label-name	ALL LABELS
address	ALL FUNC
+/- offset	ALL SQL [subtype]
address-expression	ALL DLI [subtype]
CSECT name	ALL MQ [subtype]
	ALL variable

You can specify a valid location list separated by spaces or commas, or a range; e.g., statement-number THRU statement-number. THRU is a required keyword when specifying a range. A range is not valid for program-name.

---

## data

A variable, data-name, data-item, data-label, or register (*Rn*).

---

## CSR

The cursor. When valid, put the cursor on a data-item, variable, data-label, or register and press an assigned PF key.

---

## keyword

Any valid Help topic. For example: AFTER, BATCHCON, S0C1.

---

## literal

Signed or unsigned integer or floating point number, alphanumeric string (' '), hexadecimal string (X' '), or figurative constant (spaces, zeros, high-values, or low-values).

---

## conditional-expression

Any valid expression supported by Xpediter/TSO that compares the relationship of two items, tests the truth value of a specified condition, or checks when a program variable changes value. Subsequent action of the program is dependent on the result. The following standard operators are valid:

Equal (=)	Not equal (NOT =)
Greater than (>)	Not greater than (NOT >)
Less than (<)	Not less than (NOT <)
Numeric	Not numeric

---

## PF Key Settings

To change any PF key assignment, execute the KEYS command or use the SET *PFm* command.

Key	Default	Your Setting
PF1	HELP	
PF2	PEEK CSR	
PF3	END	
PF4	EXIT	
PF5	FIND	
PF6	LOCATE *	
PF7	UP	
PF8	DOWN	
PF9	GO 1	
PF10	LEFT	
PF11	RIGHT	
PF12	GO	
PF13	HELP	
PF14	FIND CSR	
PF15	END	

Key	Default	Your Setting
PF16	EXIT	
PF17	FIND IND	
PF18	LOCATE *	
PF19	UP	
PF20	DOWN	
PF21	GO 1	
PF22	DLEFT	
PF23	DRIGHT	
PF24	GO	

---

## Primary Commands

---

### AA SNAP

Displays the Abend-AID Snapshot report.

▶▶AA SNAP▶▶

---

### ACCEPT

For COBOL, assigns a value to a data item.

▶▶ACCEPT—*identifier*—FROM—*ddname*▶▶

---

### AFTER

Sets a breakpoint after the execution of an instruction.

▶▶AFTER—*location*▶▶  

AFT  
A

Valid locations could be: statement-number, paragraph-name, program-name, label-name, procedure-name, module-name, address, address-expression, CSECT

name, +/- offset, ALL STATE, ALL PARA, ALL PROG, ALL PROC, ALL LABELS, ALL FUNC, ALL SQL [subtype], ALL DLI [subtype], ALL MQ [subtype], ALL *variable*, a list separated by spaces or commas, or a range using the keyword THRU.

---

## ALLOCATE

Dynamically accesses the file allocation utility (FAU) from the debugging session, or allocates an existing file allocation list contained in the named dataset.

```

▶▶ ALLOCATE _____ ▶▶
   |_____|
   |_____| dsname

```

---

## AT

Sets a breakpoint in a program without source.

```

▶▶ AT _____ ▶▶
   |_____|
   |_____| address
   |_____|
   |_____| CSECT-name
   |_____|
   |_____| offset

```

---

## BEFORE

Sets a breakpoint before the execution of an instruction.

```

▶▶ BEFORE _____ ▶▶
   |_____| location
   |_____|
   |_____|
   |_____|

```

Valid locations could be: statement-number, paragraph-name, program-name, label-name, procedure-name, module-name, address, address-expression, CSECT name, +/- offset, ALL STATE, ALL PARA, ALL PROG, ALL PROC, ALL LABELS, ALL FUNC, ALL SQL [subtype], ALL DLI [subtype], ALL MQ [subtype], ALL *variable*, a list separated by spaces or commas, or a range using the keyword THRU.



Valid locations could be: statement-number, paragraph-name, program-name, label-name, procedure-name, module-name, address, address-expression, CSECT name, +/- offset, ALL STATE, ALL PARA, ALL PROG, ALL PROC, ALL LABELS, ALL FUNC, ALL SQL [subtype], ALL DLI [subtype], ALL MQ [subtype], ALL *variable*, a list separated by spaces or commas, or a range using the keyword THRU.

---

## COVER

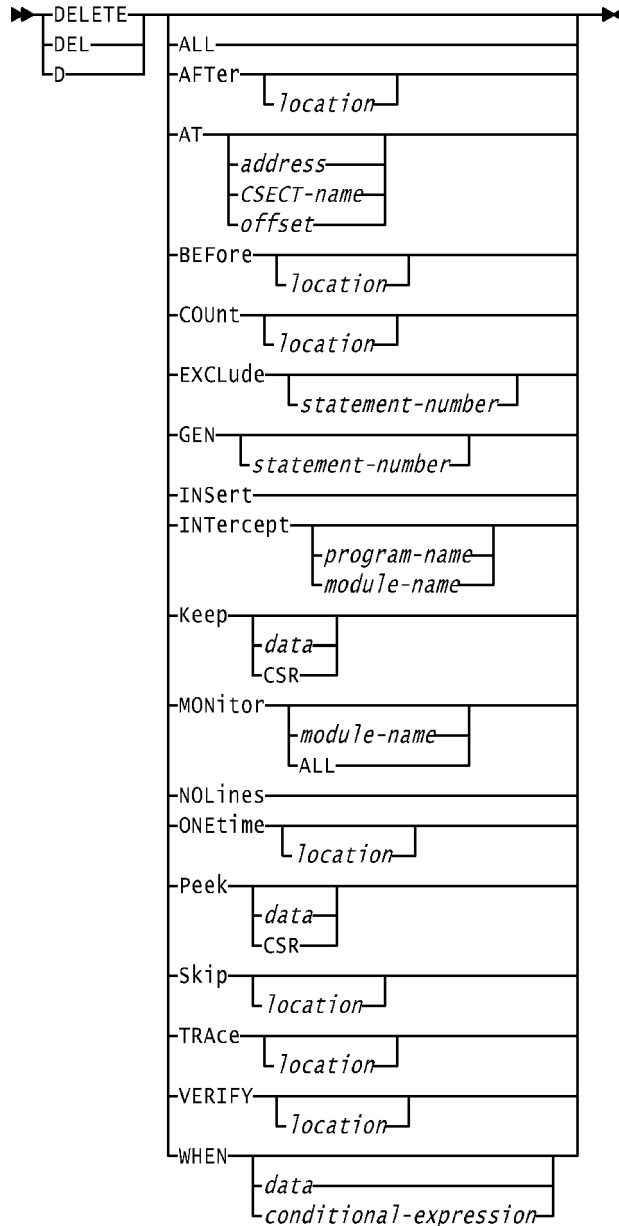
Used in the input command stream (XPIN DD) to activate Code Coverage functionality for the current test. Use the *OptionsBlock* positional parameter to specify the *System-Name* in positions 1-15 (followed by a blank), the *Test-ID* in positions 17-31 (followed by a blank), the optional *User-ID* in positions 33-40 (followed by a blank), and the *Function-ID* in position 42.

▶▶COVER—*OptionsBlock*—▶▶



# DELETE

Turns off or negates the effect of other Xpediter commands. Will also remove any unused inserted lines created with the I (Insert) line command.



Refer to the specific primary command for a list of valid *location* choices.

**Note:** GEN is valid only for COBOL and Assembler, MONITOR is valid only for COBOL, and VERIFY is valid only for Assembler.

---

## DISC

Disconnects the terminal from an Xpediter/TSO Batch Connect job.

▶▶—DISC—▶▶

---

## DLEFT

Scrolls the data in a Keep or Peek window to the left by the specified scroll amount.

▶▶—DLEFT—▶▶

<i>n</i>
Data
Half
Max
Page

---

## DLI

Issues calls to IMS using the standard IBM CBLTDLI (COBOL), ASMTDLI (Assembler), DL/I (PL/I or C), or AIBTDLI interface.

▶▶—DLI—*function-code*—

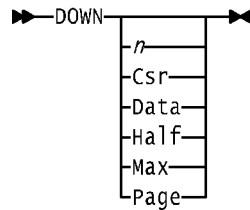
<i>PCB-n</i>
IOPCB
<i>ALTPCB-n</i>
<i>dbname-n</i>
<i>aib</i>

|—*parameters*—▶▶

---

## DOWN

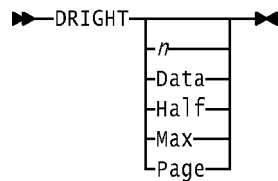
Scrolls toward the bottom of the data.




---

## DRIGHT

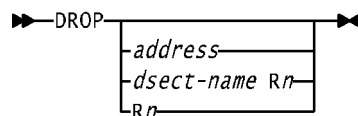
Scrolls the data in a Keep or Peek window to the right by the specified scroll amount.




---

## DROP

Valid only for Assembler, DROP releases addressability from a DSECT established with the USING command.




---

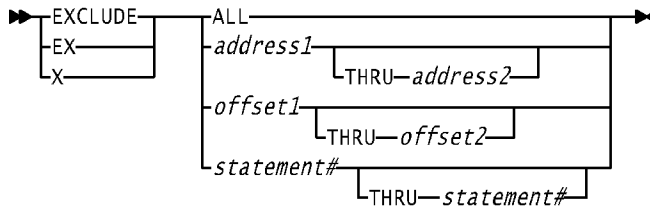
## END

Terminates the current function and returns to the previous screen.



## EXCLUDE

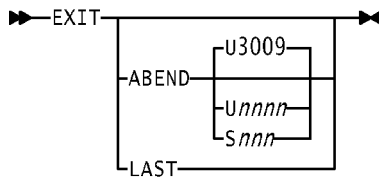
Excludes data lines from displaying in the source.



**Note:** Variables *address* and *offset* are valid only for Assembler.

## EXIT

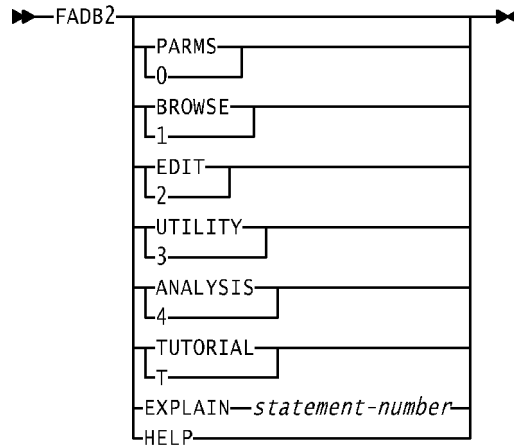
Terminates the current test session. The LAST parameter causes the step completion code to reflect the last abend condition intercepted during the session. The ABEND parameter causes the step to terminate with the specified abend code.



---

## FADB2

Accesses File-AID for DB2. Valid only with Xpediter for DB2 Extension.



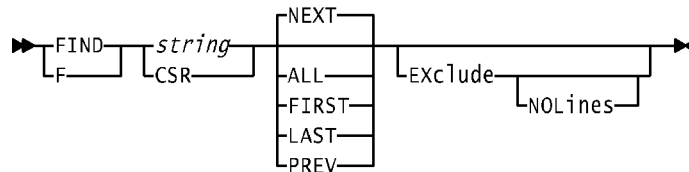
**Note:** EXPLAIN is valid only for COBOL and PL/I.

---

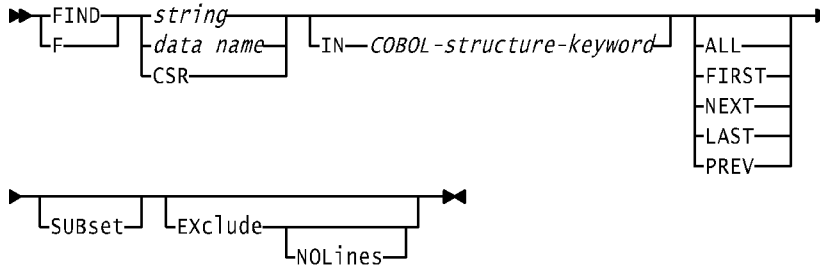
## FIND

Searches for character strings, data names, and COBOL structures. String delimiters can be "", "", == ==, or nothing. Any number of optional parameters can be specified with a required parameter. FIND without a keyword repeats the last find.

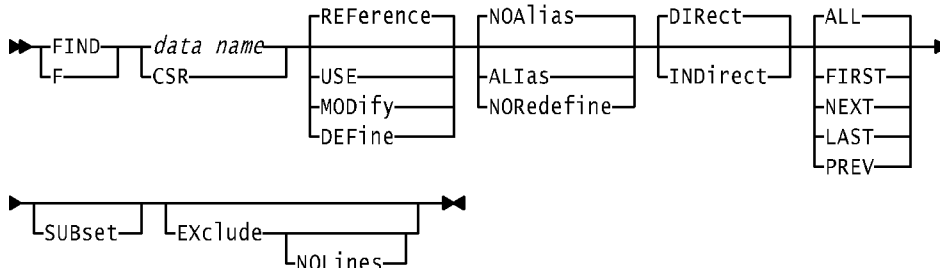
### Find Syntax (Assembler, PL/I, and C)



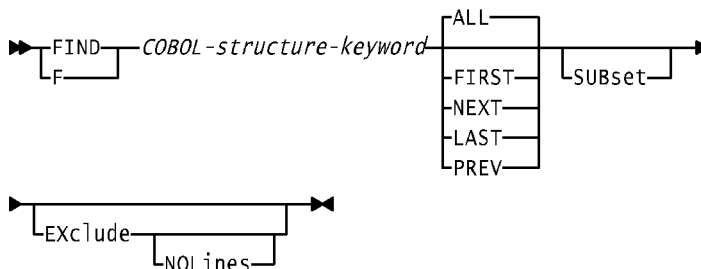
### COBOL FIND Syntax (Format 1)



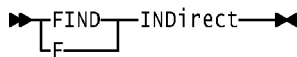
### COBOL FIND Syntax (Format 2)



### COBOL FIND Syntax (Format 3)



### COBOL FIND Syntax (Format 4)



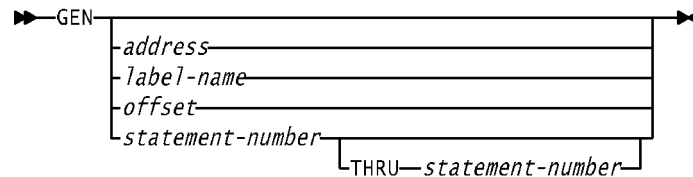
The following keywords can be used with the *COBOL-structure* parameter:

<b>ALTER</b>	Modify the value of a data name or index
<b>BRANCH</b>	Transfer logic control
<b>CALL</b>	CALL and CANCEL statements
<b>CICS</b>	EXEC CICS statements
<b>CONDition</b>	Conditional logic
<b>DLI</b>	EXEC DLI statements or calls to CBLTDLI
<b>INPut</b>	Receive data into the program
<b>IO</b>	Input and output to a program
<b>OUTput</b>	Transmit data out of the program
<b>PARAgraph</b>	Paragraph or section labels
<b>SQL</b>	DB2 statements or EXEC SQL calls

---

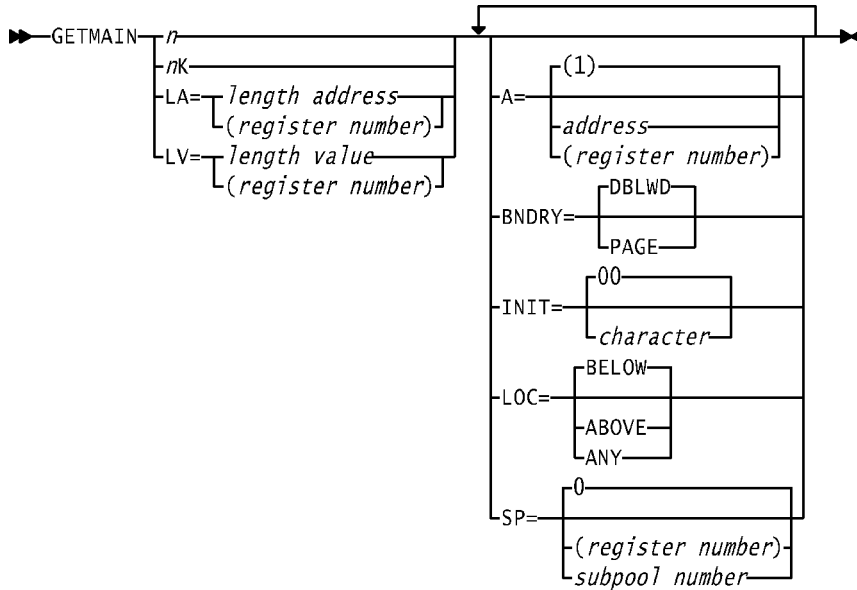
## GEN

Valid only for COBOL and Assembler, GEN expands macros or compressed EXEC code.



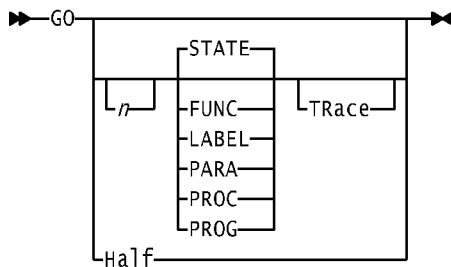
# GETMAIN

Valid only for Assembler, GETMAIN allocates virtual storage. Any number of optional parameters can be specified with a required parameter.



# GO

Begins execution or resumes execution following a pause.





---

## GOBACK

Valid only for COBOL, GOBACK changes the program logic and returns to the next higher level module.

▶▶GOBACK▶▶

---

## GOTO

Repositions the current execution pointer.

▶▶GOTO *statement-number* ▶▶  
 | *line-number* |  
 | *paragraph-name* |  
 | *label-name* |  
 | *address* |  
 | *offset* |

### CAUTION:

Be careful using the GOTO command. When you GOTO a source statement, you can also be inadvertently bypassing execution of several underlying machine instructions. This can affect subsequent statements and cause unpredictable results including, but not limited to, abends (especially S0C4 and S0C7), loops, storage overlays, and logic and display errors.

---

## GPREGS

Displays the contents of the general-purpose registers.

▶▶GPREGS 

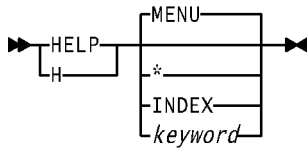
ON
OFF
LOG

 ▶▶

---

## HELP

Displays additional information about an error message or provides tutorial information. HELP \* redisplay the most recently browsed help file.




---

## IF

Establishes a conditional expression within a block of inserted lines.

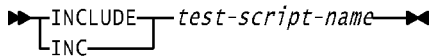
```

IF conditional-expression
.
.
.
Xpediter/TSO commands
.
.
.
[ELSE
.
.
.
Xpediter/TSO commands]
END-IF
  
```

---

## INCLUDE

Executes a predefined test script during the session.




---

## INSERT

Temporarily inserts Xpediter/TSO commands. The primary command syntax outlined below can be used only in unattended batch or in an INCLUDE test script. INSERT is only valid as a line command in interactive mode. You can remove all unused inserted lines by entering a DELETE or RESET primary command with no keywords.

```

INSert statement-number (or label-name for Assembler)
.
.
.
Any number of Xpediter/TSO commands
.
.
.
END-INS

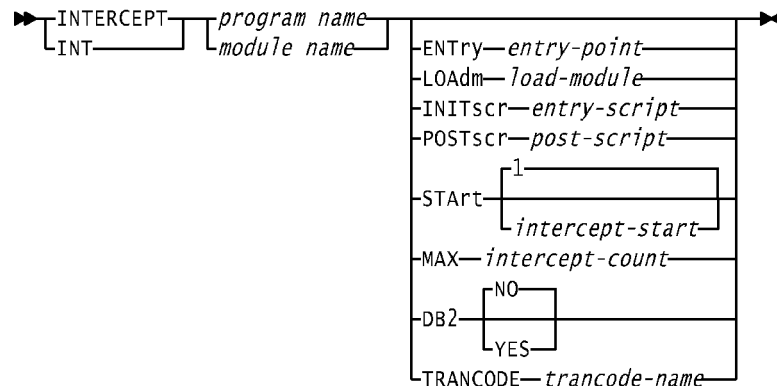
```

---

## INTERCEPT

In an interactive test, the INTERCEPT command loads a module, sets before and after breakpoints, and displays the source. In an interactive test in BTS or Dialog Manager, Xpediter/TSO automatically generates the command.

In an unattended batch test, INTERCEPT is used in the command stream to identify programs to be intercepted. In a BTS batch test, the only commands allowed in the command stream are INTERCEPT command parameters and the XPED command.



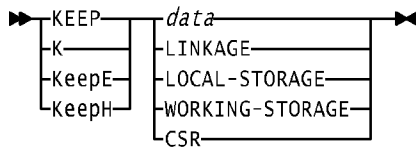
**Note:** ENTRY is not valid for PL/I. Assembler can use only ENTRY and LOADm.

---

## KEEP

Continuously displays the values of program variables in a Keep window. KeepE keeps the contents of the elementary items of a group level variable. KeepH keeps

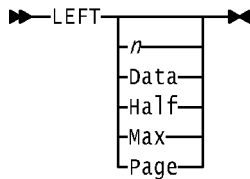
the contents in hexadecimal format. Displayed values are updated as each breakpoint is encountered.



**Note:** KeepE is not valid for Assembler. LINKAGE, LOCAL-STORAGE, and WORKING-STORAGE are valid only for COBOL. Pointer qualification (->) is valid for PL/I.

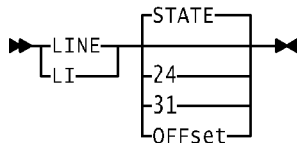
## LEFT

Scrolls the source listing to the left by the specified scroll amount.



## LINE

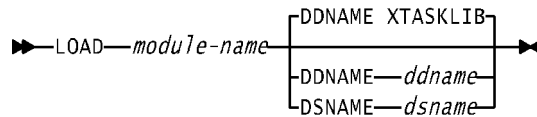
Determines whether the line command area will display statements, offsets, or addresses.



---

## LOAD

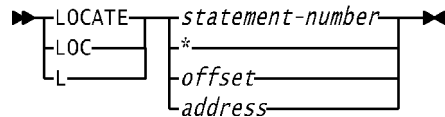
Dynamically loads a module.




---

## LOCATE

Scrolls to a particular line number in the current program. LOCATE \* scrolls to the current execution point.




---

## LOG

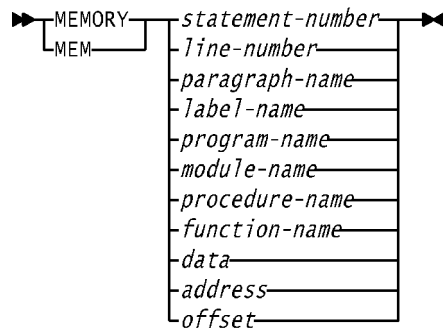
Browses the session log.




---

## MEMORY

Displays memory from a specified location.

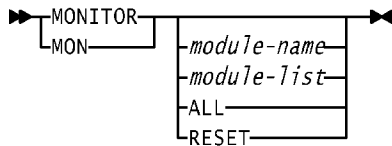


Pointer qualification (->) is valid for PL/I.

---

## MONITOR

Valid COBOL, MONITOR records the program's execution in a buffer. REVERSE can be used to execute in review mode at a later time.

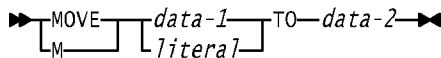



---

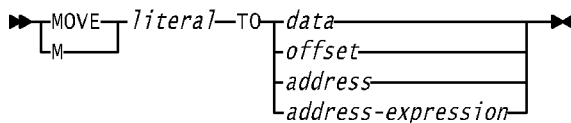
## MOVE

Changes the contents of program variables. In an interactive test, typing over the displayed value is an implicit MOVE command.

### COBOL, PL/I, and C Syntax



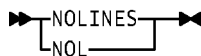
### Assembler Syntax




---

## NOLINES

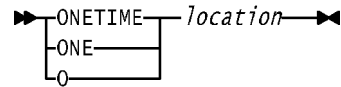
Suppresses the message **XXX LINES NOT DISPLAYED** that appears after an EXCLUDE command or keyword is issued.



---

## ONETIME

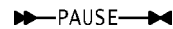
Sets a temporary before breakpoint to pause once before execution of an instruction.




---

## PAUSE

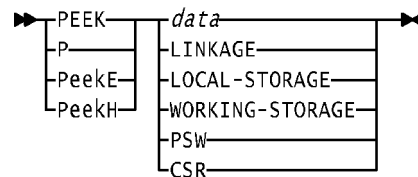
Sets a pause breakpoint within a block of inserted lines (interactive mode) or commands (batch mode).




---

## PEEK

Displays the values of program variables. PeekE displays the contents of the elementary items of a group level variable. PeekH displays the contents in hexadecimal format.

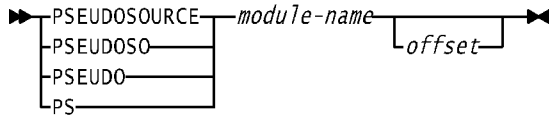


**Note:** PeekE is not valid for Assembler. LINKAGE, LOCAL-STORAGE, and WORKING-STORAGE are valid only for COBOL. Pointer qualification (->) is valid for PL/I.

---

## PSEUDOSOURCE

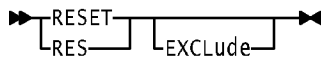
Creates a temporary pseudo-assembler view for a module or CSECT that has no matching Compuware source listing.




---

## RESET

Restores excluded lines on the source screen, removes any pending line commands, and removes any unused inserted lines created with the I (Insert) line command.




---

## RESUME

Valid only for COBOL, RESUME exits review mode and returns to the current execution position.




---

## RETEST

Begins a new test of the same program, even if the current test has not completed. Valid only in the TSO standard environment.





---

## RETURN

Valid only for PL/I and Assembler, RETURN changes the program logic and goes to the return linkage of a program.

▶▶—RETURN—◀◀

---

## REVERSE

Valid only for COBOL, REVERSE reviews the execution path that led to the current breakpoint. Requires previous entry of the MONITOR command.

▶▶—REVERSE—▶◀  
    └—REV—┘

---

## RIGHT

Scrolls the source listing to the right by the specified scroll amount.

▶▶—RIGHT—▶◀  
    ┌—*n*—┐  
    ├—Data—┤  
    ├—Half—┤  
    ├—Max—┤  
    └—Page—┘

---

## RUN

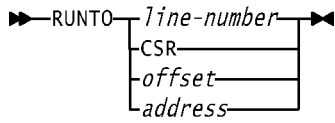
Submits an execution JCL file and connects to the specified job.

▶▶—RUN—*dsname*—▶◀

---

## RUNTO

Sets a onetime breakpoint and starts execution.



---

## SCRNSAVE

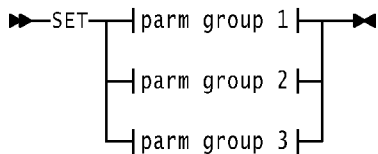
Copies the current Xpediter/TSO screen to the Xpediter session log.



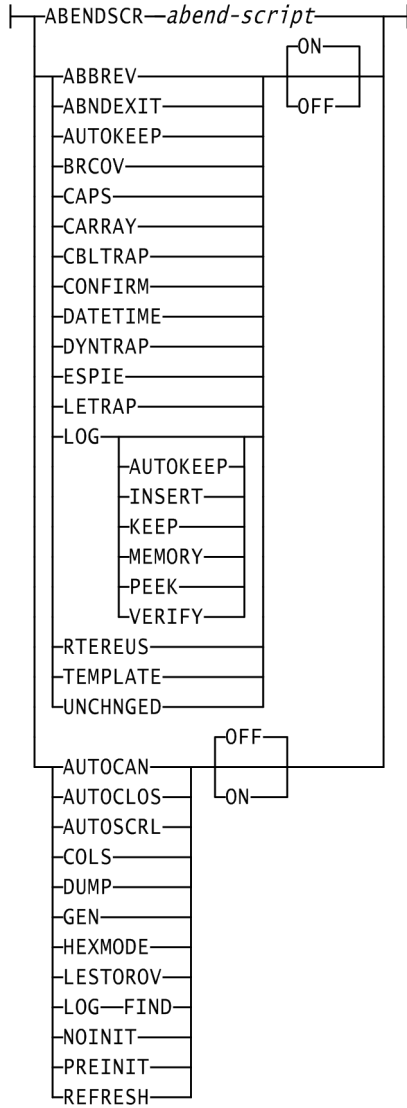
---

## SET

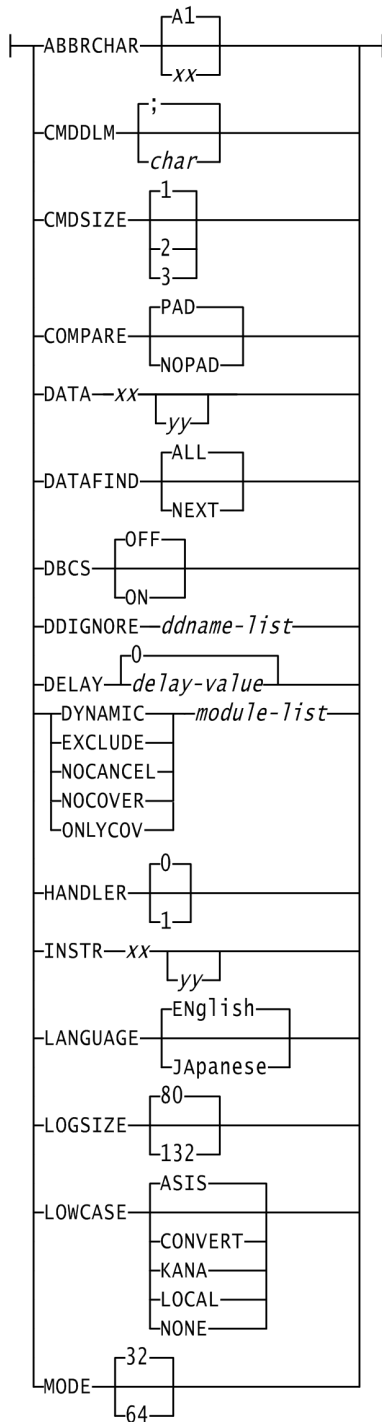
Overrides Xpediter/TSO defaults. Some values are set only for the duration of the test session, while others are maintained across sessions. For more information, see the *Xpediter/TSO and Xpediter/IMS Reference Manual*.



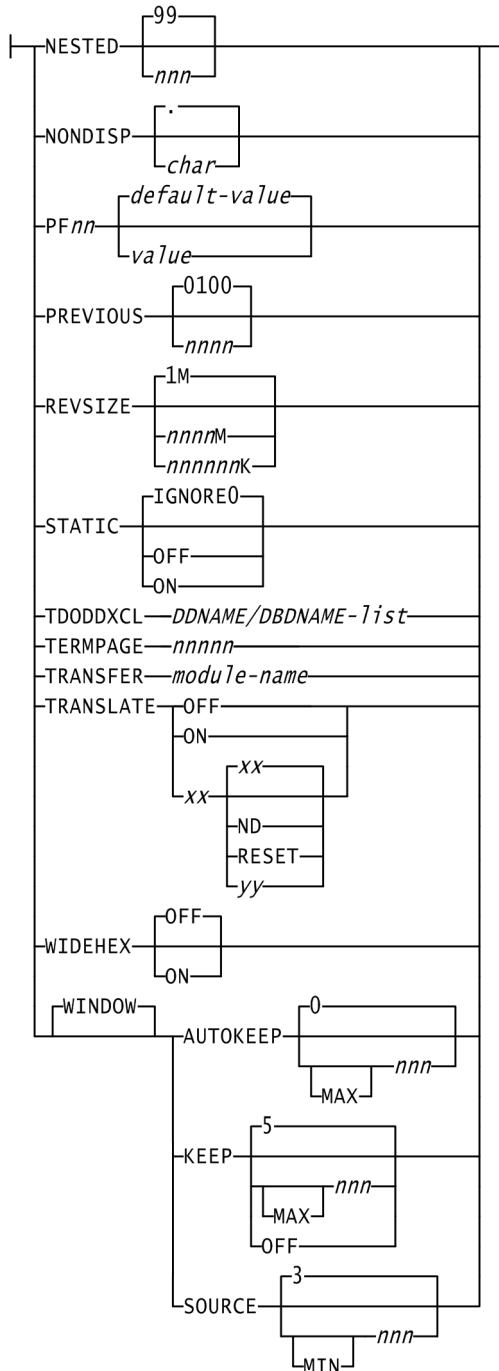
# Parameter Group 1



## Parameter Group 2



## Parameter Group 3

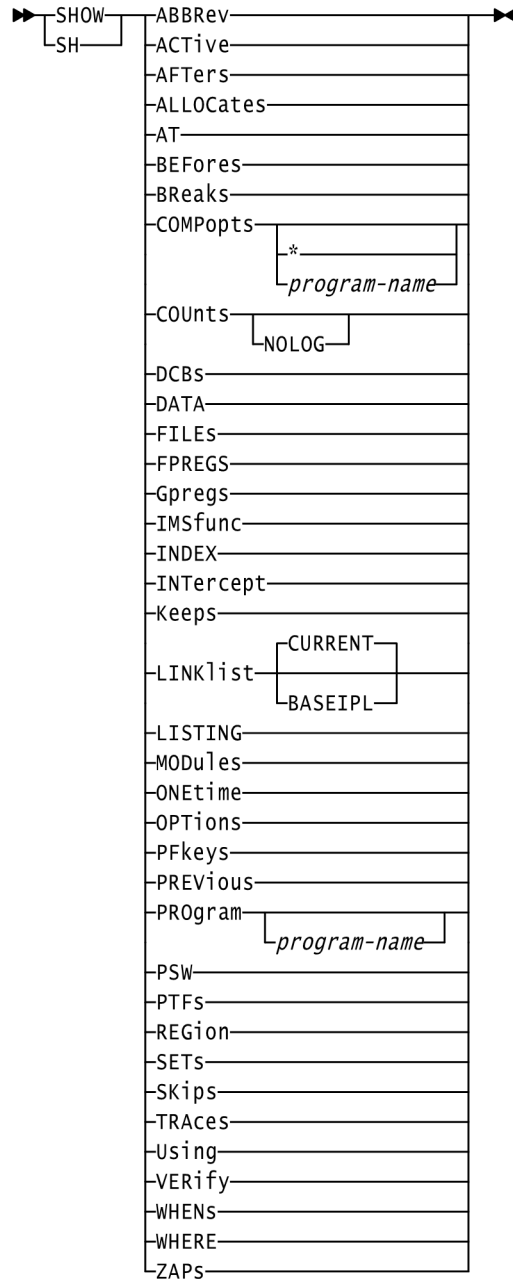


The following restrictions apply to certain SET command parameters:

Parameter	Restriction
ABENDSCR	Valid only in an unattended batch test
AUTOCAN	COBOL only
CARRAY	C language only
CBLTRAP	COBOL only
CMDDL	Valid only in a Batch Connect test
DATA xx yy	Assembler only
DATAFIND	COBOL only
DELAY	Not valid in an unattended batch test
DYNTRAP	COBOL only
GEN	Assembler and COBOL only
INSTR xx yy	Assembler only
LOG VERIFY	Assembler only
NOCANCEL	COBOL only
NOINIT	COBOL only
REVSIZ	COBOL only
RTEREUS	COBOL only

# SHOW

Displays breakpoints, diagnostic information, or the SET command options.

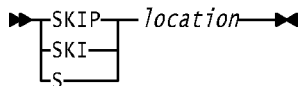


**Note:** AFTER, AT, BEFORE, BREAK, LISTING, ONETIME, PFKEY, and SKIP parameters are not valid in unattended batch. COMPOPT, DCB, FILE, and INDEX are valid only for COBOL. USING and VERIFY are valid only for Assembler.

---

## SKIP

Temporarily bypasses the execution of a statement.



Valid locations could be: statement-number, paragraph-name, program-name, procedure-name, label-name, address, +/- offset, address-expression, module-name, ALL STATE, ALL PARA, ALL PROC, ALL LABELS, ALL PROG, ALL FUNC, ALL SQL [subtype], ALL DLI [subtype], ALL MQ [subtype], ALL *variable*, a list separated by spaces or commas, or a range using the keyword THRU.

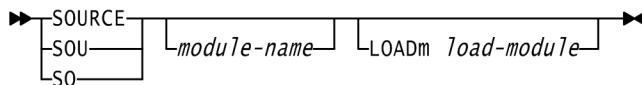
### CAUTION:

**Be careful using the SKIP command. When you SKIP a single source statement, you can also be inadvertently bypassing execution of several underlying machine instructions. This can affect subsequent statements and cause unpredictable results including, but not limited to, abends (especially S0C4 and S0C7), loops, storage overlays, and logic and display errors.**

---

## SOURCE

Changes the module shown on the source display when testing interactively. In unattended batch (or in an INCLUDE file), use the SOURCE command to qualify a module to be associated with subsequent Xpediter commands.



The parameter descriptions for the SOURCE command are:

### *module-name*

The name of the module, external procedure, entry point, or CSECT to be displayed on the Source screen. You can also refer to a program or module by the name of the load module.





---

## TOP

Scrolls to the top of the data.

▶▶TOP◀◀

---

## TRACE

Monitors the program's logic flow. If MAX n is not used, the default is 25 in interactive mode and 2500 in unattended batch mode.

▶▶TRACE *location* ◀◀  
 ↳TRA ↳MAX *n*

Valid locations could be: statement-number, paragraph-name, program-name, procedure-name, label-name, address, +/- offset, address-expression, module-name, ALL STATE, ALL PARA, ALL PROC, ALL LABELS, ALL PROG, ALL FUNC, ALL SQL [subtype], ALL DLI [subtype], ALL MQ [subtype], ALL *variable*, a list separated by spaces or commas, or a range using the keyword THRU.

---

## UP

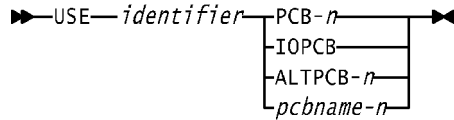
Scrolls toward the top of the data.

▶▶UP◀◀  
 ↳*n*  
 ↳Csr  
 ↳Data  
 ↳Half  
 ↳Max  
 ↳Page

---

## USE

Valid only for COBOL in IMS subprogram testing. Establishes addressability for database PCBs in the linkage section.

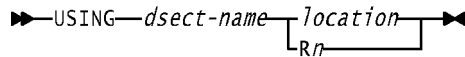


Valid with Xpediter/TSO only when the value of SUB is specified as a test session parameter for the IMS environment.

---

## USING

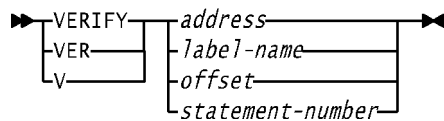
Valid only for Assembler, USING temporarily establishes addressability to a DSECT. USING is effective only if the program has established a base register for the DSECT and loaded the base register with the appropriate value.




---

## VERIFY

Valid only for Assembler, VERIFY displays the contents of an instruction or data area.

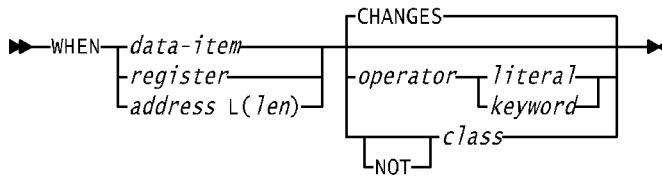



---

## WHEN

Indicates when a specified condition is true or when a program variable changes value. In an interactive test, execution is paused. In an unattended batch test,

execution does not pause. A message is written to the session log indicating that the specified condition has been met.

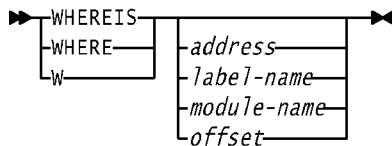


**Note:** Pointer qualification (->) is valid for PL/I.

---

## WHEREIS

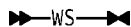
Displays the location of a specific address or CSECT.




---

## WS

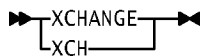
Valid only for COBOL, WS displays working storage.




---

## XCHANGE

Accesses Xpediter/Xchange to simulate date and time changes for your applications.



---

## XPED

Specifies the environment in the input command stream (XPIN DD). Valid environments are BATCH (default), BATCH PEM, BTS, BTS PEM, IMS, IMS PEM, TSO, and TSO PEM.

▶▶XPED—*environment*▶▶

---

## Line Commands

Double-character line commands are used to specify a block of lines. An **n** indicates a number.

---

### **), )n, )) , ))n**

Scrolls a line or block of displayed data one or n columns to the left.

---

### **(, (n, ((, ((n**

Scrolls a line or block of displayed data one or n columns to the right.

---

### **:**

Freezes the line at the current column.

---

### **:n**

Scrolls a line to column n.

---

### **A, AA**

Sets an after breakpoint on a line or block.

## **B, BB**

Sets a before breakpoint on a line or block.

---

## **C, CC**

Sets a count on a line or block.

---

## **D**

Deletes all breakpoints on a Procedure Division line.

Deletes the displayed value on a line.

Deletes an inserted line.

Deletes the kept value on a kept line.

Reshows all the excluded lines on an excluded range of lines.

---

## **D:**

Enables left and right scrolling on a frozen line.

---

## **DA**

Deletes the after breakpoint on a line.

---

## **DB**

Deletes the before breakpoint on a line.

---

## **DC**

Deletes the count on a line.

---

**DD**

Deletes all commands and breakpoints on a block of lines.

---

**DE**

Deletes the display of elementary values.

---

**DG**

Collapses the macro on a line.

---

**DH**

Deletes the values displayed in hexadecimal format.

---

**DO**

Deletes the onetime breakpoint on a line.

---

**DS**

Deletes the skip on a line.

---

**DT**

Deletes the column template from above a displayed variable.

---

**DV**

Deletes the verified field.

Deletes the trace on an instruction.

## E, En, EE

Displays the elementary items for the first or nth variable on a line or block.

---

## F, Fn

Reshows the first or n line(s) from a block of excluded lines.

---

## G, GG

Expands macros or compressed EXEC code (GEN) on a line or block.

---

## GT

Repositions (GOTO) the current execution pointer to a line.

**CAUTION:**

**Be careful using the GOTO command. When you GOTO a source statement, you can also be inadvertently bypassing execution of several underlying machine instructions. This can affect subsequent statements and cause unpredictable results including, but not limited to, abends (especially S0C4 and S0C7), loops, storage overlays, and logic and display errors.**

---

## H, Hn, HH

Displays the first or nth variable on a line or block in hexadecimal format.

---

## I, In

Inserts one or n line(s) below the current line.

---

## K, Kn, KK

Keeps the first or nth variable on a line or block.



---

## **KE, KEn**

Keeps the elementary items for the first or nth variable on a line.

---

## **KH, KHn**

Keeps the first or nth variable on a line in hexadecimal format.

---

## **K\***

Keeps all variables on a line.

---

## **L, Ln**

Reshows the last or n line(s) from a block of excluded lines.

---

## **M**

Displays memory starting from the address associated with a line.

---

## **O, OO**

Sets a onetime breakpoint on a line or block.

---

## **P, Pn, PP**

Temporarily displays (PEEK) the first or nth variable on a line or block.

---

## **PE, PEn**

Displays the elementary items for the first or nth variable on a line.

---

## **PH, PHn**

Displays the first or nth variable on a line in hexadecimal format.

## **P\***

Displays all variables on a line.

---

## **S, SS**

Sets a skip on a line or block.

### **CAUTION:**

**Be careful using the SKIP command. When you SKIP a single source statement, you can also be inadvertently bypassing execution of several underlying machine instructions. This can affect subsequent statements and cause unpredictable results including, but not limited to, abends (especially SOC4 and SOC7), loops, storage overlays, and logic and display errors.**

---

## **T, TT**

Displays a column template above a displayed or kept item or block.

Sets a trace on an instruction or block.

---

## **V, VV**

Verifies an instruction or data area on a line or block.

---

## **X, XX**

Excludes a line or block.

---

## **XP**

Captures and displays EXPLAIN information about an EXEC SQL or inserted SQL statement. Valid only with the Xpediter for DB2 Extension.

---

## **Z**

Sets a onetime breakpoint and starts execution.

---

## Link-Edit Options

Link-edit options OVLY and NE are not supported. Note that NOTEST deletes SYM records from the load module for Assembler.

---

## Unattended Batch Commands

Unattended batch is the processing of data without interacting with the debugging session from your terminal. Commands are read from a test script and the output from the test session is written to the session log. Use any of the following commands in a script for an unattended debugging session:

AFTER	GOBACK	LINE	RESET	USE
BEFORE	GOTO	LOAD	SET	USING
COUNT	IF	MOVE	SHOW	WHEN
DELETE	INCLUDE	ONETIME	SKIP	WS
DROP	INSERT	PAUSE	SOURCE	XPED
EXIT	INTERCEPT	PEEK	TEST	
GO	KEEP	PSEUDOSOURCE	TRACE	

