

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page v.

Second Edition (September 1994)

This publication replaces and makes obsolete the previous edition, SC09-1759-00. The changes for this edition are indicated by a vertical bar in the left margin.

This edition applies to Version 3, Release 1, Modification Level 0, of IBM Application Development ToolSet/400 (Program 5763-PW1) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada. M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "Communicating Your Comments to IBM" for a description of the methods. This page immediately precedes the Readers' Comment Form at the back of this publication.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1993, 1994. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of International Business Machines Corporation, Armonk, N.Y.

Contents

Notices	v
About This Guide	vii
How to Use This Guide	vii
Who Should Use This Guide	vii
Summary of Changes	ix
Chapter 1. Introducing the Interactive Source Debugger Tool	1
What is the Interactive Source Debugger Tool?	1
Starting the Interactive Source Debugger	2
Command Syntax	2
Parameters	2
Using Online Help	6
Restarting or Ending a Session	6
Ending a session	6
Restarting a session	6
Chapter 2. Using the Source Display	7
Introducing the Source Display	7
Using Windows and Displays	8
Using the Source Display Menu Bar	8
Using Function Keys	9
Using Commands	10
Using Equate Names	12
Using System Commands	13
Clearing Equate Names	13
Using a Predefined Set of Equate Names	13
Storing Information in a Profile	14
Logging Commands	14
Setting Logging Off	15
Moving Around in the Source	15
Using Labels	16
Finding Text	16
Using the Find String Window	17
Setting Source Display Options	18
Changing Highlighting Used in the Source Display	18
Changing Scrolling and Shifting Values	18
Setting Other Options	19
Chapter 3. Debugging Programs	21
Working with Breakpoints	21
Setting Breakpoints	21
Setting Conditional Breakpoints	22
Setting Breakpoints with SKIP Values	22
Displaying Breakpoints	23
Removing Breakpoints	23
Executing a Program	23
Running through a Program	23
Stepping through a Program	24

About Unmonitored Messages	24
Working with Variables	24
Displaying Variables	25
Displaying RPG Array Elements	25
Displaying RPG Indicators	26
Printing Variables	26
Changing Variables	27
Watching Variables	28
Removing Watched Variables from the Source Display	29
Displaying Information about a Program	30
Working with More than One Program	30
Adding a Program to the Debugging Environment	31
Changing the Program You Are Viewing	32
Removing a Program from the Program List	32
Debugging Programs in Other Jobs	32
Working with Source on Another AS/400 Machine	33
Bibliography	35
Index	37

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, Connecticut, USA 06904-2501

The following terms, denoted by an asterisk (*), used in this publication, are trademarks or service marks of International Business Machines Corporation in the United States or other countries:

- Application System/400
- AS/400
- COBOL/400
- IBM
- IBMLink
- Integrated Language Environment
- ILE
- PROFS
- RPG/400
- 400

About This Guide

This guide describes how to use the interactive source debugger (ISDB) component of Application Development ToolSet/400 (ADTS/400) to carry out debugging on the Application System/400* (AS/400*) system.

It contains procedural information on:

- What ISDB does
- How to start and stop the tool
- Using the Source display to view program source and perform actions on it
- How to set breakpoints and display variables.

How to Use This Guide

This guide contains examples to help you do common tasks. The displays illustrated in these examples may differ from your own because the names of the AS/400 libraries, objects, and members may be different from yours.

Command names are usually shown in full, in uppercase. Where they are shown in a combination of upper- and lowercase characters, the uppercase letters designate a short form for the command. For example, for the command 'Break', you can enter either BREAK or B. Refer to "Using Commands" on page 10 for more information.

You can enter characters on the command line and in prompts, in uppercase, lowercase, or mixed case.

This book does not contain a glossary. New terms, when introduced, are highlighted in **boldface** type.

You may need to refer to other IBM manuals for more specific information about a particular topic. For a list of related publications, see the "Bibliography" on page 35. (For information on all the manuals in the AS/400 library, see the *Publications Guide*, SC41-3003.)

Who Should Use This Guide

This guide is intended for application programmers or analysts working in an AS/400 environment. To use it effectively, you must be familiar with the AS/400 operating system and one of the following programming languages:

- COBOL/400*
- Control Language (CL) Version 3 Release 1
- RPG/400* Version 3 Release 1
- System/36-Compatible RPG II
- System/38 Environment Option of the RPG Compiler

Summary of Changes

The following enhancements have been made to the interactive source debugger tool:

- COBOL/400, from Version 2 Release 2 or later, is now supported for source-level debugging.
- A new method of invoking and ending ISDB is implemented. You can now call a program at the same time you start ISDB, and when you end a program, ISDB also ends. (This feature is not available for serviced jobs, which still must use the invocation and ending methods for Version 2.)
- You can now start debugging programs again without having to stop and restart ISDB. Your debugging environment (such as watched variables and set break-points) is maintained.
- You can now start ISDB from option 34 of the Programming Development Manager (PDM) utility. For more information on how to use this option, see the *ADTS/400: Programming Development Manager* book, SC09-1771.

Chapter 1. Introducing the Interactive Source Debugger Tool

This chapter describes the interactive source debugger (ISDB) tool, and explains how to start and stop it.

What is the Interactive Source Debugger Tool?

The interactive source debugger (ISDB) tool provides an interactive environment for using the AS/400 system debugger to debug the following types of programs:

- Control Language (CL) Version 3 Release 1
- COBOL/400 compiled with the *SRCDBG option
- RPG/400*, System/36-Compatible RPG II, and the System/38 Environment Option of the RPG Compiler

Note that Integrated Language Environment* (ILE*) programs are **not** supported.

You can use this tool to:

- Debug up to ten related CL, COBOL, or RPG programs.
- View the source of a program in debug mode, so that you can see the source code as you are debugging it.
- Move around in the source, find text strings, and mark places with labels so that you can return to them later.
- Perform actions using either menu bars and pull-down windows, the command line, or function keys.
- Set up to 50 breakpoints for each program you are debugging, including conditional breakpoints and breakpoints that stop execution of the program at a statement only after the statement has been encountered a specified number of times.
- Display, monitor, and change the values of variables.
- Execute a program until it encounters a breakpoint or a specified line, or step through the program line by line.
- Log the commands you have used, for later retrieval.
- Debug programs running in other jobs.

Note that you cannot use ISDB to debug listing views of programs, or to edit the source of a program you are viewing in the ISDB Source display, since it is locked.

Also, you cannot use the **Source display** command line to call other programs that you want to debug using ISDB. (To debug them, you must exit from the Source display, and then call the new program from the AS/400 command line.)

Starting the Interactive Source Debugger

Figure 1 shows the syntax for the STRISDB command that starts the interactive source debugger. A description of the command's parameters follows.

Command Syntax

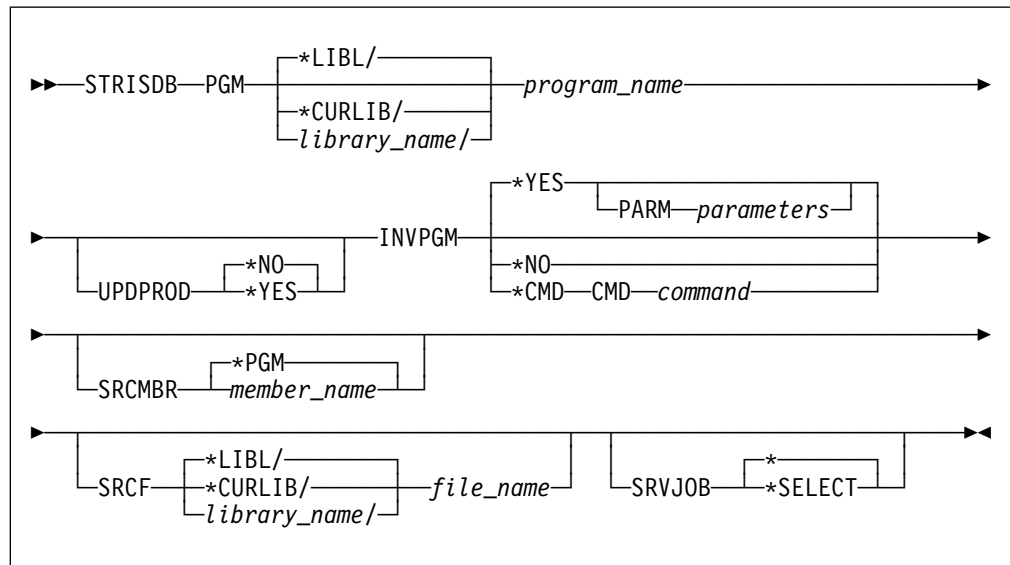


Figure 1. STRISDB Command Syntax

Parameters

Program (PGM)

Use this parameter to specify the library and name of the program to be debugged.

The possible values for library names are:

***LIBL**

The library list is used to locate the program. This is the default if no library name is specified.

***CURLIB**

The current library in the library list is used to locate the program. (If you have not specified a current library, QGPL is assumed.)

library-name

Specify the name of the library that contains the program you want to debug.

Note: To debug COBOL programs with ISDB, ensure that they are compiled with the *SRCDDBG option.

Update Production Files (UPDPROD)

Use this parameter to specify whether files in production libraries can be changed while they are in debug mode.

The possible values are:

***NO**

Files cannot be updated while they are in debug mode.

***YES**

Files can be updated while they are in debug mode.

Note that you can change the update production value from the Source display. To do this:

1. Select the *Debug* menu-bar choice from the Source display.
2. In the Debug pull-down window, select option 1 (Change debug). The Set Debugging Options window appears.
3. Select either option 1 (Yes) to update production files, or option 2 (No) if you do not want to update them, then press Enter.

Invoke Program (INVPGM)

Use this parameter to specify whether you want ISDB to call your program, do other preparation before you invoke your program, or use a command to invoke your program.

***YES**

ISDB calls the program specified in the PGM parameter. If the program requires parameters, specify them in the PARM parameter.

Note: You cannot use this parameter for serviced jobs.

***NO**

A command is not issued to start the program. A command entry screen (QCMD) is provided so that you can do any preparation required before you issue the command to start the program.

If you start your program from the command entry screen, when your program completes running, control will return to this screen. To End ISDB or Restart, you must first exit this command entry screen (F3 or F12). The Program Termination display will then appear where you can choose to end ISDB or restart.

Note: For servicing other jobs, this is the only option you can use, but the QCMD panel is not provided.

***CMD**

ISDB executes the command specified in the CMD parameter. (This command must invoke your program.)

Parameters (PARM)

Use this parameter to specify any character string parameters required to call your program. If you wish to use a numeric parameter, enter the CALL command with the required parameters, in the CMD option.

This parameter is not intended for the INVPGM(*NO) and INVPGM(*CMD) invocation commands.

Command (CMD)

Use this parameter to specify the command you want to use to invoke your program.

This parameter is not intended for the INVPGM(*NO) and INVPGM(*YES) invocation commands.

Source Member (SRCMBR)

Only specify this parameter if the source member you want to work with is different from the one specified in the object description of the program. When you invoke the program, the source of the member you specify here will be displayed.

You will need to use this parameter if:

- The library, file, or member names of the source have changed since the program was last compiled. (Be sure the member contains the correct program source, or you will get unpredictable results.)
- The program is an RPG Auto Report program. The CRTRPTPGM command has a parameter that lets you place the expanded source into a source physical file.
- The source is located on another AS/400 machine and its program was not created using a DDM file.

The possible values are:

***PGM**

The source information is retrieved from the object description using the DSPOBJD command.

member_name

Specify the name of the source member you want displayed.

Source File and Library (SRCF)

Use these parameters to specify the library name, and file name of the source member you specified in the *Source member* prompt. You must specify both names if either one of them is different from the names specified in the object description of the program.

The possible values for the source library name are:

***LIBL**

The library list is used to locate the source file.

***CURLIB**

The current library is used to locate the source file. (If you have not specified a current library, QGPL is assumed.)

library_name

Specify the name of the library that contains the source file you want displayed.

The possible value for the source file name is:

file_name

Specify the name of the file that contains the source member you want displayed.

Job to Service (SRVJOB)

Use this parameter if you want to debug a program running in a job different from the one to which you are signed on. It is useful for debugging batch jobs or other interactive jobs.

Possible values are:

* Debug in the current job.

*SELECT

Display the Select Job to Service display, which lists all active jobs. When you select one of these jobs, the STRSRVJOB command is issued and that job is placed in debug mode.

Note: SRVJOB(*SELECT) is not valid with INVPGM(*YES) or INVPGM(*CMD).

Examples of the STRISDB command

```
STRISDB RTVMSGCL
```

Debugs the CL program RTVMSGCL with no special parameters.

```
STRISDB PGM(LABELS) INVPGM(*YES) PARM(CUST)
```

Debugs the program LABELS, and invokes it with the command CALL PGM(LABELS) PARM(CUST).

```
STRISDB PGM(MYPROG) INVPGM(*CMD) CMD(CALL MYPROG PARM(1 2))
```

Debugs the program MYPROG which requires numeric parameters. Therefore, it must be invoked with a CALL command.

```
STRISDB REPORT INVPGM(*CMD) CMD(STRAPPL PRINT)
```

Debugs the program REPORT and invokes it with the user-defined command STRAPPL PRINT.

```
STRISDB PGM(AUTO) SRCMBR(AUTO) SRCF(USERLIB/SAMPLES)
```

Debugs the RPG program AUTO that contains auto report specifications. The source member AUTO is obtained through using the command CRTRPTPGM PGM(AUTO) SRCFILE(QIXA/SAMPLES) OUTFILE(USERLIB/SAMPLES) OUTMBR(AUTO)

Once you issue the STRISDB command with the appropriate parameters, the Source display appears, with the program stopped at the first statement that can be executed.

If your program is in a serviced job, after you issue the STRISDB command, you must then invoke it from the command line (for example, with the CALL command).

For more information on the Source display, see Chapter 2, "Using the Source Display" on page 7.

Using Online Help

In addition to the help provided in this User's Guide, you can use the online help available with the interactive source debugger tool.

To use the online help:

- Press F1 when the cursor is located on a prompt for detailed help on that particular prompt.
- Select the *Help* menu-bar choice in the Source display, and then select one of the options listed there for detailed help on function keys or commands.
- Press F2 from some help displays for extended help on the window or display you are using.

Restarting or Ending a Session

Ending a session

For both serviced and non-serviced jobs, you can end an ISDB session either when you are stopped in your program, or when it has completed.

When stopped in your program:

- Press F3 or F12, or issue the QUIT or QQUIT commands from the Source display. If the Cancel ISDB window appears, select option 1 (Yes).
- For serviced jobs only, you return to the command line where you must issue the ENDISDB command, which deletes any work files or spool files that may have been created by ISDB, and unlocks source so that you can update it. There are no parameters for this command.

When your program has completed:

- For non-serviced jobs, the Program Termination display appears. Choose option 1 (End ISDB).
- For serviced jobs only, you return to the command line where you must issue the ENDISDB command, which deletes any work files or spool files that may have been created by ISDB, and unlocks source so that you can update it.

Restarting a session

You can only restart ISDB for non-serviced jobs:

- To restart ISDB when your program has stopped, issue the RESTART command from the Source display
- To restart it after your program has completed, wait until the Program Termination display appears, and choose option 2 (Restart program).

Note: To repeat the actions you performed in the previous session, copy the member containing the log to another member so that you can use it as input for the next session. For details on how to do this, see “Logging Commands” on page 14.

Chapter 2. Using the Source Display

This chapter describes how to use the Source display to view the source of a program you are debugging.

It explains:

- What the Source display is
- What you can do with the menu bar, pull-down windows, function keys, and command line
- How to use equate names to customize how you enter commands
- How to save your settings so that they are invoked every time you start a debugging session
- How to use a command log to repeat actions in another debugging session
- How to view places in the source
- How to find a desired string of text in the source
- How to change how the display appears and behaves

Introducing the Source Display

When you start the ISDB tool and invoke a program, the Source display appears. From this display you can add and remove breakpoints, display program variables, and perform many other debugging functions.

Debug	Goto	Program	Options	Help
LRSLIB/TSTPGM: /209000 ISDB/400 More: - + >				
199000	C	PRCSFL BEGSR		
200000	C	READ FILE	99	
201000	C	*IN99 DOWEQOFF		
202000	C	Z-ADDAMT	TEMP	
203000	C	TEMP MULT 100	TEMP00	
204000	C	TOTAL ADD TEMP00	TOTAL	
205000	C	WRITESUBFILE		
206000	C	ADD 1	COUNT	
207000	C	COUNT IFGT 100		
208000	C	Z-ADD0	TEMP00	
209000	C	Z-ADD0	TOTAL	
210000	C	Z-ADD1	COUNT	
211000	C	EXCPTOUT		
212000	C	END		
213000	C	READ FILE		99
214000	C	FLAG COMP '*'	23	
215000	C	*IN23 IFEQ '1'		
216000	C	MOVE 'DONE'	TEMP01	

ISDB ==>>

F3=Exit F5=Step F6=Break F11=Display variable
F12=Cancel F17=Run F23=Change variable F24=More keys

Stopped at statement 209000.

Figure 2. Source Display Containing RPG/400 Source

The following features are available on the Source display:

- The first line is the **menu bar**, which gives you quick access to the product's features. See "Using the Source Display Menu Bar" on page 8.

- The second line is the **status line**, showing the name of the program and the number of the statement that is currently being executed.
- The fourth-last line (preceded by "ISDB ==>") is the **command line**, where you can enter commands.
- The two lines following the command line show the function keys you can use to perform functions quickly. To view other sets of keys, press F24. See "Using Function Keys" on page 9.
- The last line is the **message line**, where product or system messages are displayed.
- The rest of the display contains the source of the program you are debugging. The currently executing line, if visible, is highlighted.

To change how this screen appears and behaves, see "Setting Source Display Options" on page 18.

Using Windows and Displays

You can access various other windows and displays from the Source display, using either the menu bar, or by issuing a command (for example: DSP *RPGIND).

Sometimes, you may want to view a display again and again as you are debugging a program. To view the display you last viewed, press F15 or enter the SCREEN command. When you are finished, press Enter. This feature is not supported for 132-column displays.

Using the Source Display Menu Bar

Several **pull-down windows** are provided on the Source display. These windows are available from the menu-bar choices (*Debug*, *Goto*, *Program*, *Options*, and *Help*), and contain functions that you will use frequently.

To use a function listed in a pull-down window:

1. Press F10 to put the cursor on the menu bar, and then use the tab key to move to the choice you want.
2. Press Enter. The pull-down window for that choice is displayed.
3. To select an option, either type its number, or position the cursor on it, and press Enter.
4. If an ellipsis (...) follows an option, then another display or window will be displayed to prompt you for additional information.

Alternatively, you can view a pull-down window by typing its name on the command line and pressing Enter. (For the Goto window, use the command GO.)

If an option is not available, it is displayed in blue (on a color terminal), or normal intensity (on a monochrome terminal), and nothing happens if you select it.

Summary of Menu-Bar Choices

The following menu-bar choices are available from the Source display:

- Debug** Lets you set debugging options, display breakpoint or debug information, clear all breakpoints, or work with the list of programs you are debugging. For more information, see Chapter 3, “Debugging Programs” on page 21.
- Goto** Lets you view another place in the program source. For more information, see “Moving Around in the Source” on page 15.
- Program** Lets you view and change various attributes of the program you are debugging, including the job it is running on. For more information, see “Displaying Information about a Program” on page 30.
- Options** Lets you set ISDB options, such as how much the Source display should scroll. (For more information, see “Setting Source Display Options” on page 18.) You can also invoke the Find String window to find text or set search options. (For more information, see “Finding Text” on page 16.)
- Help** Provides access to online help displays.

Using Function Keys

Table 1 lists the function keys used in the Source display. For further details on any of these keys, go to the appropriate section in this book, or use the online help.

Notes:

1. Some keys are sensitive to the location of the cursor. For example, you cannot use F6 (to set a breakpoint) unless the cursor is on a valid statement.
2. If you position the cursor on a variable and press F11 or F22 to display it, or F23 to change it, it is not recognized if its entire name is not visible on the Source display.

Table 1 (Page 1 of 2). Summary of Source Display Function Keys

Function Key	Description	For more information...
F1	Get Help	“Using Online Help” on page 6
F3	Exit	“Restarting or Ending a Session” on page 6
F4	Go to currently executing line	“Moving Around in the Source” on page 15
F5	Step	“Executing a Program” on page 23
F6	Add/Remove Breakpoint	“Working with Breakpoints” on page 21
F7	Page Up	“Moving Around in the Source” on page 15
F8	Page Down	“Moving Around in the Source” on page 15
F9	Retrieve previous command	“Using Commands” on page 10
F10	Go to menu bar	“Using the Source Display Menu Bar” on page 8
F11	Display variable	“Working with Variables” on page 24

Table 1 (Page 2 of 2). Summary of Source Display Function Keys

Function Key	Description	For more information...
F12	Cancel	"Restarting or Ending a Session" on page 6
F13	Run to location	"Executing a Program" on page 23
F14	Display program list	"Working with More than One Program" on page 30
F15	Display last screen you used	"Using Windows and Displays" on page 8
F16	Repeat find	"Finding Text" on page 16
F17	Run	"Executing a Program" on page 23
F19	Left	"Moving Around in the Source" on page 15
F20	Right	"Moving Around in the Source" on page 15
F21	System command	"Using Commands" on page 10
F22	Display variable in hexadecimal	"Working with Variables" on page 24
F23	Change variable	"Working with Variables" on page 24
F24	More keys	"Introducing the Source Display" on page 7

Using Commands

The **command line** on the Source display accepts commands recognized by the interactive source debugger, or commands that you have associated with an EQUATE command (see "Using Equate Names" on page 12).

To retrieve a command you used previously, press F9. Up to ten commands are kept at a time.

To enter an AS/400 system command, precede it with SYS (or SYS ? if you require prompting). Alternatively, press F21 to use the Command Entry window.

Table 2 lists the ISDB commands.

Table 2 (Page 1 of 3). Source Display Command Summary

Command	Description	For More Information...
AddisdbPgm	Add a program to the program list	"Working with More than One Program" on page 30
BACKward	Page backward	"Moving Around in the Source" on page 15
BOTtom	Go to end of program source	"Moving Around in the Source" on page 15
Break	Set breakpoints	"Working with Breakpoints" on page 21
Chg	Change a variable	"Changing Variables" on page 27
ChgisdbPgm	Change program	"Working with More than One Program" on page 30
CLear	Clear breakpoints	"Removing Breakpoints" on page 23

Table 2 (Page 2 of 3). Source Display Command Summary

Command	Description	For More Information...
DEbug	Display Debug pull-down window	"Using the Source Display Menu Bar" on page 8
Down	Roll program down <i>nnn</i> lines	"Moving Around in the Source" on page 15
Dsp	Display a variable	"Working with Variables" on page 24
DspHex	Display a variable in hexadecimal	"Working with Variables" on page 24
EQuate	Assign an equate name to a command	"Using Equate Names" on page 12
Find	Find a string	"Finding Text" on page 16
FORward	Roll program down <i>nnn</i> pages	"Moving Around in the Source" on page 15
GO	Display Goto pull-down window	"Using the Source Display Menu Bar" on page 8
Goto	Go to a position in the program	"Moving Around in the Source" on page 15
LaBeL	Set a label	"Using Labels" on page 16
Left	View information to the left of the display	"Moving Around in the Source" on page 15
nnnn	Go to a statement number	"Moving Around in the Source" on page 15
Options	Display Options pull-down window	"Using the Source Display Menu Bar" on page 8
PRogram	Display Program pull-down window	"Using the Source Display Menu Bar" on page 8
Prt	Print a variable	"Working with Variables" on page 24
PrtHex	Print a variable in hexadecimal	"Working with Variables" on page 24
Quit	Quit ISDB with confirmation	"Restarting or Ending a Session" on page 6
QQuit	Quit ISDB without confirmation	"Restarting or Ending a Session" on page 6
ReStArt	Reinvoke a program to debug it again	"Restarting or Ending a Session" on page 6
RFind	Repeat find	"Finding Text" on page 16
Right	View information to the right of the display	"Moving Around in the Source" on page 15
RmvisdbPgm	Remove a program from the program list	"Working with More than One Program" on page 30
RUN	Run the program	"Executing a Program" on page 23
SCReen	Display previous screen	"Using Windows and Displays" on page 8
Set Break	Set breakpoint display	"Setting Source Display Options" on page 18
Set COLOR	Set colors in the Source display	"Setting Source Display Options" on page 18

Table 2 (Page 3 of 3). Source Display Command Summary

Command	Description	For More Information...
Set CURsor	Set where cursor is positioned after a step operation	"Setting Source Display Options" on page 18
Set FLOW	Display RPG program flow arrows	"Setting Source Display Options" on page 18
Set LOG	Keep a log	"Logging Commands" on page 14
Set Roll	Set how the program rolls	"Setting Source Display Options" on page 18
STEP	Step through the program	"Executing a Program" on page 23
SYS	Issue a system command	The introduction to this table.
Top	Go to the top of the program	"Moving Around in the Source" on page 15
UnWatch	Stop monitoring variables	"Working with Variables" on page 24
UP	Roll program up <i>nnn</i> lines	"Moving Around in the Source" on page 15
USE	Use a command file	"Using Equate Names" on page 12; "Logging Commands" on page 14
Watch	Monitor a variable	"Working with Variables" on page 24
WatchHex	Monitor a variable in hexadecimal	"Working with Variables" on page 24

Notes on Commands

- The command names in Table 2 are shown in a combination of upper- and lowercase characters. The uppercase characters are required, and can be used alone as a short form for the command. For example, you can enter either "RIGHT" or just "R."
- For complete information on commands, see the online help.
- It is not necessary to put quotes around statement numbers or variable names.
- It is not necessary to include an ampersand (&) in CL variable names, except for the WATCH, WATCHHEX, and UNWATCH commands.
- The parameters for AS/400 debug commands (such as ADDBKP and RMBKP) are not checked, and the Source display is not updated to reflect the commands' actions.

Using Equate Names

If you use a command frequently, you may choose to assign a shorter **equate name** to it. The equate name must be assigned to an entire command; it will not work for just part of a command (such as a long variable name).

An equate name will override any other commands with the same name.

Syntax:

```
Equate name <definition>
```

Examples

```
EQUATE CLRALL CLEAR *ALL
```

Assigns the name CLRALL to the command that removes all breakpoints.

```
EQ B BOTTOM
```

Assigns the name B to the command that moves the cursor to the bottom of the source. Note that the default command for B is BREAK, but it will be ignored.

Using System Commands

You can assign a name to a system command by preceding the command with SYS.

Examples

```
EQUATE WS SYS WRKSPLF
```

Assigns the name WS to the system command for working with spool files.

```
EQ CE SYS CALL QCMD
```

Assigns the name CE to the system command that brings up the command entry screen.

Clearing Equate Names

To remove an equate name, enter the name without its definition.

Examples

```
EQUATE CLRALL  
EQ WS
```

The above examples remove the equate definitions CLRALL and WS, respectively.

```
EQ *CLR
```

The above example shows the use of the special value *CLR for the equate name to clear all equate definitions.

Using a Predefined Set of Equate Names

To save a set of equate commands for use in other debugging sessions, store them in a source member. Precede any comment lines you create in the profile with an asterisk (*) or a slash and an asterisk (/*) as the first characters. To then invoke the equate commands, type USE, followed by the library name, file name, and member name of this source member, on the Source display command line.

Storing Information in a Profile

If you want certain settings (such as equate commands or SET commands) to be used every time you start the interactive source debugger tool, include them in a source member and store it in a file called QIXAPROF, which is in your library list (*LIBL/QIXAPROF).

Every time you start ISDB, it looks in your library list for the QIXAPROF file, and uses this member as a **profile**. A profile contains settings that are run automatically when the Source window is opened.

A sample profile is included in the member QIXA/SAMPLES(PROFILE). Copy this member to the QIXAPROF file and modify it to your needs.

Logging Commands

By default, typed commands or function keys that affect how you are debugging the program are logged in a source member, QIXALOG, which is stored in the source physical file QIXALOG in the library QTEMP.

To use this log to reproduce your actions in another debugging session:

1. Exit your interactive source debugger session as usual.

All commands you typed on the command line, and all other actions that affected the execution of the program, are stored in the file QIXALOG. (Commands that were run from a file with the USE command are also logged, although the USE command itself is commented out.)
2. Since the log file may contain commands that are not valid, you may want to edit it before using it again so that you have a clean set of instructions.
3. Copy the QTEMP/QIXALOG file to a file with a different name. (This file is cleared the next time you use the STRISDB command, so you must copy it.)
4. Enter the STRISDB command to start another debugging session for the same program.
5. On the Source display command line, type USE followed by the name of the source physical file you copied QIXALOG to.

Example

```
USE QTEMP TEMP QIXALOG
```

where TEMP is the name of the file to which you copied QIXALOG.

The commands you used in the previous session are now run again.

Note: If you enter the USE command to execute a number of commands and the program ends before the last command is processed, the Program Termination window will appear. You can then either quit, or start debugging the program again. If you start debugging the program again, ISDB will discard the rest of the commands. (For serviced jobs, you must end ISDB by issuing the ENDISDB command before debugging another program, to guarantee predictable results.)

Setting Logging Off

Logging is turned on by default. To turn it off, type SET LOG OFF. To turn it on again, type SET LOG ON.

Moving Around in the Source

There are various methods you can use to move around in the source of the program that you are debugging.

The Goto pull-down window provides one method to change your view of the program.

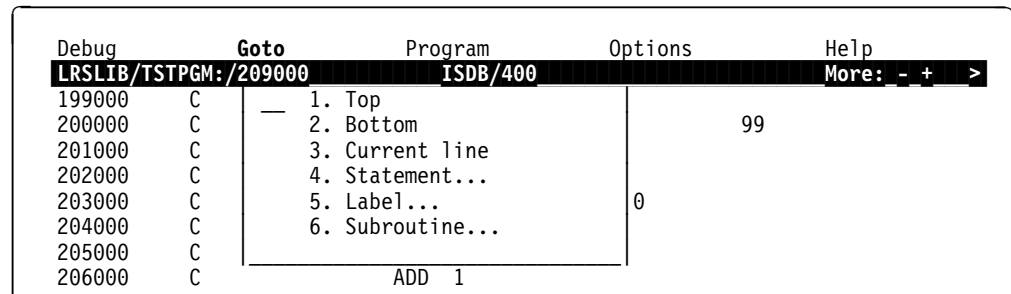


Figure 3. Goto Pull-Down Window

- To position the source so that you see its first line, select option 1 (Top). (Alternatively, type TOP or GOTO TOP.)
- To position the source so that you see its final line, select option 2 (Bottom). (Alternatively, type BOTTOM or GOTO BOTTOM.)
- To position the source so that you see the currently executing line, select option 3 (Current line). (Alternatively, type GOTO CURRENT.)
- To use a window where you can enter the number of the line you want to see, select option 4 (Statement). (Alternatively, type the line number on the command line, or type the line number after the GOTO STMT command, and press Enter.)
- To use a window that contains all of the labels you have associated with places in the program and enables you to choose the label you want to go to, select option 5 (Label). (For more information on labels, see “Using Labels” on page 16.)
- To use the Goto Subroutine window, where you can enter the name of the RPG subroutine you want to see, select option 6 (Subroutine). (Alternatively, use the GOTO BEGSR command.)

In addition to the choices in the Goto pull-down window, you can use the following commands and function keys to move around in a program.

- To scroll one screen backward, press the Page Up or F7 key, or type BACKWARD.
- To scroll several lines backward, type UP followed by the number of lines you want to scroll.

- To scroll one screen forward, press the F8 or Page Down key, or type FORWARD.
- To scroll several lines forward, type DOWN followed by the number of lines you want to scroll.
- To view the contents to the right of the screen, press F20 or type RIGHT. To view the contents to the left again, press F19 or type LEFT.

(For information on how to change the number of columns that are shifted or lines that are scrolled, see “Changing Scrolling and Shifting Values” on page 18.)

Using Labels

To mark a place in the program you are debugging so that you can return to it quickly, you can set a **label**. Up to eight labels can be set per program.

To set a label:

1. Labels are set at the line of the program that is currently displayed as the top line of source in the Source display. Scroll through your program so that the location where you want to set a label is the top line in the Source display.
2. Type LABEL, followed by the command with the name you want to assign to the label.

To return to the location in a program where you have set a label:

- On the command line, type GOTO LABEL, followed by the name of the label, and press Enter.

If you have forgotten the names of the labels you set, select the *Goto* menu-bar choice, and then *Label* (or type GOTO LABEL without parameters). The Goto a Label window appears, listing all the labels you have set. Select the name of the label you want to go to.

You cannot remove labels, but if you set more than eight, the oldest one is replaced with the newest one you set.

Finding Text

To find a particular string of text in the source, do one of the following:

- Select the *Options* menu-bar choice, and then select option 2 (Find). The Find String window appears, where you can use prompts to enter the text you want to find and the search criteria you want used.
- Press F16. A repeat search (with the same search options) is done for the text you found previously. (If no text has yet been found, the Find String display appears.) To find a new string of text, type the relevant text on the command line before you press F16.
- Use the FIND command. If you do not follow the command with a string of text, the Find String display appears to prompt you. If you do follow the command with a string of text, a search is made for that string.
- Use the RFOUND command to find the next occurrence of text that was found previously. The search direction is the same as that of the previous search.

Using the Find String Window

To use the Find String window:

1. Select the *Options* menu-bar choice, and then select option 2 (Find). (Alternatively, use the FIND command without parameters.) The Find String window is displayed.

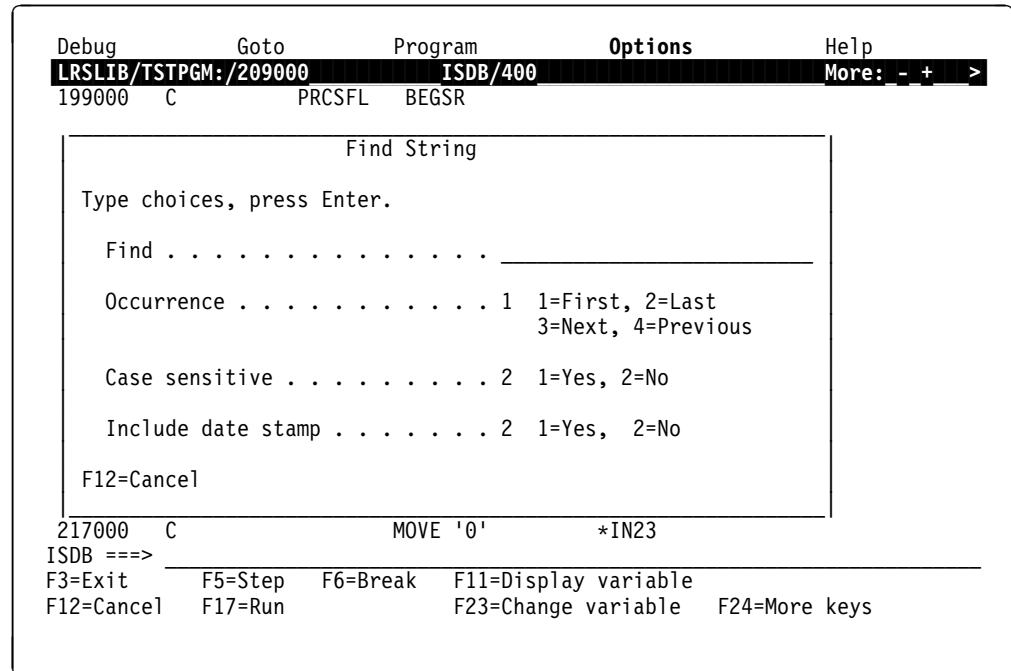


Figure 4. Find String Window

2. In the *Find* prompt, type the string you want to find.
3. In the *Occurrence* prompt, specify how you want the search to be performed.

The possible values are:

1=First

Finds the first occurrence of the string by starting at the first line in the program and searching forward.

2=Last

Finds the last occurrence of the string by starting at the last line in the program and searching backward.

3=Next

Finds the next occurrence of the string. The search starts at the line containing the last find. (If there has been no previous find, the search begins with the first record in the program and searches forward.)

4=Previous

Finds the previous occurrence of the string. The search starts at the line containing the last find. (If there has been no previous find, the search begins with the last record in the program and searches backward.)

4. If you want the search to be case-sensitive so that only exact matches of upper- and lowercase characters are found, type 1 in the *Case sensitive* prompt. If you do not want this, type 2.

5. If you also want the date stamp for the string to be searched, type 1 in the *Include date stamp* prompt. If you do not want this, type 2.
6. Press Enter. You are returned to the Source display, with the cursor positioned at the search string if it was found. If it was not found, a message appears at the bottom of the display.

Setting Source Display Options

You can change various settings to make the Source display appear or behave differently.

To find out how to save these settings so that they are invoked automatically when you start ISDB, see “Storing Information in a Profile” on page 14.

Changing Highlighting Used in the Source Display

Table 3 shows the default colors and highlighting used in the Source display.

Table 3. Default Highlighting in the Source Display

Item	Default Highlighting: Monochrome Terminal	Default Highlighting: Color Terminal
Currently executing line	Reverse image	Red
Line in which a text string was found during a search	High intensity	White
Line number requested	High intensity	White
Line with breakpoint	Reverse image	Reverse image

To change the default highlighting, use the SET COLOR command. (Refer to the online help for details on how to use it.)

Changing Scrolling and Shifting Values

You can modify how many lines or columns are moved in the Source display when you:

- Shift the display horizontally by pressing F19 (Left) or F20 (Right), or by using the LEFT or RIGHT commands.
- Scroll vertically in the program by pressing the F7, Page Up, F8, or Page Down, or by using the BACKWARD or FORWARD commands.

To change the default number of columns or lines moved:

1. Select the *Options* menu-bar choice.
2. Select option 1 (Set display options). The Set Display Options window appears.

```

Debug      Goto      Program      Options      Help
LRSLIB/TSTPGM:/209000  ISDB/400  More: - + >
199000 C      PRCSFL  BEGSR      1_ 1. Set display options...
200000 C      READ FIL  2. Find...
201000 C      *IN99    DOWEQOFF
202000
203000
204000
205000
206000
207000
208000
209000
210000
211000
212000
213000
214000          FLAG    COMP '*'          23
215000          *IN23   IFEQ '1'
216000          MOVE  'DONE'   TEMP01
ISDB ===>
F3=Exit      F5=Step    F6=Break   F11=Display variable
F12=Cancel   F17=Run    F23=Change variable  F24=More keys

```

Figure 5. Set Display Options Window

- In the *Amount to roll* prompt, set the number of lines you want scrolled. The possible values are:

H= Half

The screen is scrolled half a page (9 lines).

F= Full

The screen is scrolled a complete page (18 lines). (This is the default.)

C= Cursor

The screen is scrolled so that the line at the cursor becomes the first line on the screen.

1-999

The screen is scrolled by the number of lines you specify.

(Alternatively, you can use the SET ROLL command to set scrolling values. Refer to the online help for details.)

- In the *Amount to shift* prompt, specify the number of positions you want the display to shift to the right or to the left.

Setting Other Options

You can use the following SET commands to set other options:

Set Break Long | Short

Specifies whether the entire line of set breakpoints is highlighted (**Long**), or only their statement numbers (**Short**).

Set CURsor CoMmand | Follow

Specifies whether the cursor is positioned on the command line, or on the line that is currently executing when you press F5 to step through the program line by line.

Set FLOW oN | ofF

Specifies whether arrows indicating RPG program flow are displayed or not.

The arrows show situations where the flow of an RPG program changes, and will be displayed when the program is going into operations such as TAG or BEGSR, or when it is going out of operations such as GOTO or ENDSR.

The arrows are shown at the leftmost columns of the display, and will not be visible if you shift the display to the right.

Chapter 3. Debugging Programs

This chapter describes how to debug programs using ISDB.

It explains how to:

- Set breakpoints (including conditional breakpoints), display the ones you set, and remove them
- Run and step through a program
- Display, monitor, change, and print values of variables and RPG array elements
- Show which RPG indicators are turned on
- Display information on the status of a program
- Work with more than one of the programs related to the program you are debugging, so that you can debug a program if it is called
- Debug programs in other jobs
- Work with the source of a program on another AS/400 machine

Working with Breakpoints

You can set up to 50 breakpoints for each program you are debugging. A **breakpoint** is a place in a program where you can interrupt its running so that you can take actions (such as tracking variables) to try to debug it.

Setting Breakpoints

To set a breakpoint, position the cursor on the line where you want the breakpoint set, and press F6. (A breakpoint is set at the first statement in the line for COBOL programs.)

To set several breakpoints, or to set a breakpoint at a particular statement in the line, use the BREAK command, following it by up to five statement numbers. For COBOL programs, you can specify a particular statement by following its line number with a period and its ordinal.

Examples

```
BREAK 11300.3
```

For COBOL, stops at the third statement in line 11300. (This command with a period and an ordinal is not valid for other languages.)

```
BREAK 100 200 300
```

Stops at lines 100, 200 and 300. (For COBOL, breakpoints are set only at the first statements in these lines.)

Setting Conditional Breakpoints

You can set a **conditional breakpoint** which will cause the program to stop running on a line only if certain conditions are met.

Syntax:

```
Break stmt WHEN var condition value
```

Adds a conditional breakpoint at the statement *stmt*. The program stops if the expression *var condition value* is met.

Possible values of *condition* are:

*EQ (or =)	*NE (or ≠)	*GT (or >)
*LT (or <)	*GE (or ≥)	*NL (or ≮)
*NG (or ¬>)	*CT	

Examples

```
BREAK 11300 WHEN COUNT = 10
```

Stops at line 11300 if the variable *count* is equal to 10. (For COBOL, the program stops at the first statement in the line.)

```
BREAK 12700.2 WHEN TEMP > 30
```

For COBOL, stops at the second statement in line 12700 if the variable *temp* is greater than 30.

```
BREAK 660 WHEN TOTAL *CT SUB
```

Stops at line 660 if the variable *TOTAL* contains the character string contained in the variable *SUB*. (For COBOL, the program stops at the first statement in the line.)

Setting Breakpoints with SKIP Values

You may want the program to stop at a statement only after that statement has been run a certain number of times. This feature is useful, for example, to examine a section of a program that goes into a loop.

Syntax:

```
Break stmt SKIP nnn
```

Stops at the statement *stmt* after it has been run *nnn* times.

Example

```
BREAK 1200 SKIP 10
```

The program stops at the statement in line 1200 if the statement has been run ten times. (For COBOL, the first statement in the line is assumed.)

Displaying Breakpoints

To see a list of the breakpoints you have set:

1. Open the Debug pull-down window and select option 2 (Display breakpoints). The Display Breakpoints display appears, listing the breakpoints you have set for the program you are currently debugging. You can scroll through this list by pressing F7 and F8, or the Page Down and Page Up keys.

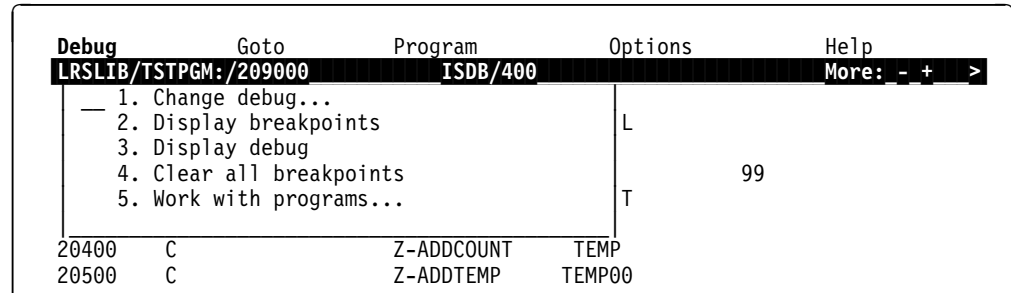


Figure 6. Debug Pull-Down Window

2. Press F3 or F12 to return to the Source display.

Removing Breakpoints

To remove a breakpoint that you no longer want to use, position the cursor in the Source display on the line where it is set, and press F6. For COBOL programs, only the first breakpoint in the line is removed.

To remove up to five breakpoints from the command line, use the CLEAR command followed by the statement numbers of up to five breakpoints.

To remove the breakpoint from a COBOL statement, specify its line number followed by a period and the ordinal of the statement.

To remove all breakpoints you have set in a program, do one of the following:

- Use the CLEAR * command on the command line
- Open the Debug pull-down window and select option 4 (Clear all breakpoints).

Executing a Program

Once you have set the breakpoints you want, you can start running the program. You can either run it until a breakpoint is reached, or step through it by running only one or more of its statements at a time.

Running through a Program

To run through the program until a breakpoint is reached, press F17, or use the RUN command.

To run through the program until a particular statement is reached, position the cursor on that statement in the Source window and press F13, or use the RUN command followed by its statement number. The program will run until that statement is reached.

If you start running a program and it does not encounter the statement to which you requested it to run, it will continue running to completion. If this occurs, use the Program Termination display that appears to either quit or start debugging the program again. For serviced jobs, you must use the ENDISDB command before debugging another program, to guarantee predictable results.

Stepping through a Program

There may be sections of the program that you will want to debug in detail. You can use ISDB to step through the program line by line, so that you can check the values of variables after each statement.

To step through a program one statement at a time, press F5, or use the STEP command without any parameters.

To run through several statements of the program before stopping it, use the STEP command followed by the number of statements you want the program to run.

Notes:

1. When debugging RPG programs, you cannot step through /COPY members or externally described files.
2. If you use the STEP command to step through a number of statements and the program ends before the last step operation is executed, the Program Termination display will appear. You can then quit or start debugging the program again.
3. For serviced jobs, you must use the ENDISDB command before debugging another program, to guarantee predictable results.

About Unmonitored Messages

If the debugger encounters an error in a program that produces an unmonitored AS/400 message, it stops running the program and displays the message on the Source display message line. The line where the error was encountered is highlighted in the source (if the program where it was encountered is being debugged).

Working with Variables

You can use ISDB to keep track of variables when a program stops at a breakpoint.

Usage Notes:

- For the descriptions in the following sections, note that you can use substrings to work with a particular part of the contents of a variable. To work with substrings, use the following syntax:

(start<,length>)varname

where *start* is the first position of the character of the substring, *length* is its length, and *varname* is the name of the variable. This does not apply to the WATCH commands. For more information, refer to the examples in this section or to the online help.

- For COBOL variable names used by more than one data structure, you can use the IN or OF keywords. For more information on COBOL usage, consult the online help.

Displaying Variables

While debugging a program, you may want to view the values of variables when the program stops at a breakpoint.

To view the contents of a variable at any time:

1. Type DSP on the command line, followed by the names of up to five variables that you want to view. (If you want to view hexadecimal values, type DSPHEX, followed by the names of up to five variables.)
2. Press Enter. The Display Program Variables display appears, showing the format, type, and length of the first variable.
3. Press Enter. If you specified more than one variable for the DSP command, the Display Program Variables display appears for each variable you specified. Press Enter when you are finished viewing each display. When you press Enter after viewing the display for the final variable, you are returned to the Source display.

Alternatively, position the cursor on a variable name that is visible on the Source display and press F11 (or F22 if you want to view its hexadecimal value).

Examples

```
DSP temp1 temp2 temp3
```

Displays the contents of the variables temp1, temp2, and temp3.

```
DSPHEX (80)address
```

Displays the contents of the variable address in hexadecimal format from the 80th character.

```
DSP 'FIELD1 OF STRUC1'
```

For COBOL, displays the contents of the variable FIELD1 of the data structure STRUC1.

```
DSP (50,10)VAR1
```

Displays 10 characters of the variable VAR1, starting from the 50th character.

Displaying RPG Array Elements

To display an RPG array element:

1. Type DSP on the command line, followed by up to five array names and element numbers. Use the form *array(el)*, where *array* is the array name and *el* is the array element number.
2. Press Enter. The Display Program Variables display appears, showing the format, type, and length of the first array element.
3. Press Enter. If you specified more than one array element for the DSP command, the Display Program Variables display appears for each array element you specified. Press Enter when you are finished viewing each display. When you press Enter after viewing the display for the final array element, you are returned to the Source display.

(You can also print, change, or watch RPG array elements if you follow the instructions in the following sections and use array names and element numbers for the variable names.)

Example

```
DSP names(3) data(2)
```

Displays the contents of the the third element of the array names, and the second element of the array data.

Displaying RPG Indicators

To display RPG indicators:

1. On the command line, type the command `DSP *RPGIND`.
2. Press Enter. The Display RPG Indicators window appears, listing the RPG indicators 01 through 99 and L1 through L9. Indicators that are on are displayed in reverse image.

```

Debug          Goto          Program          Options          Help
LRSLIB/TSTPGM:/209000          ISDB/400          More: - + >
199000 C          PRCSFL BEGSR
200000 C          READ FILE          99
201000 C
202000 C          :          RPG Indicators 1 - 99, L1 - L9          :
203000 C          :
204000 C          : 01 02 03 04 05 06 07 08 09 10 :
205000 C          : 11 12 13 14 15 16 17 18 19 20 :
206000 C          : 21 22 23 24 25 26 27 28 29 30 :
207000 C          : 31 32 33 34 35 36 37 38 39 40 :
208000 C          : 41 42 43 44 45 46 47 48 49 50 :
209000 C          : 51 52 53 54 55 56 57 58 59 60 :
210000 C          : 61 62 63 64 65 66 67 68 69 70 :
211000 C          : 71 72 73 74 75 76 77 78 79 80 :
212000 C          : 81 82 83 84 85 86 87 88 89 90 :
213000 C          : 91 92 93 94 95 96 97 98 99          :
214000 C          : L1 L2 L3 L4 L5 L6 L7 L8 L9          :
215000 C          :
216000 C          : Press Enter to continue.          :
ISDB ==>          :
F3=Exit      F5=Step    F6=Break    F11=Display variable
F12=Cancel   F17=Run    F23=Change variable    F24=More keys

```

Figure 7. RPG Indicators Window

3. Press Enter to return to the Source display.

Printing Variables

You can send variable information to the AS/400 default output queue to print it.

To print the contents of variables:

1. Type `PRT` on the command line, followed by the names of up to five variables you want to print.
2. Press Enter. The information is sent to your default output queue for printing.

To print the contents of variables in hexadecimal format, use the PRTHEx command instead of PRT.

Examples

```
PRT (60)custdata
```

Prints the contents of the variable custdata starting at character 60.

```
PRTHEx name acct
```

Prints the contents of the variables name and acct in hexadecimal format.

Changing Variables

You may want to change the value of a variable in the program as you are debugging it, to test how the program runs with various input values.

Note that you cannot change the values of COBOL pointers, or the basing values of based COBOL fields.

To change the value of a variable that is visible on the Source display:

1. Position the cursor over the variable.
2. Press F23. The Change Variable window appears, containing the name of the variable.

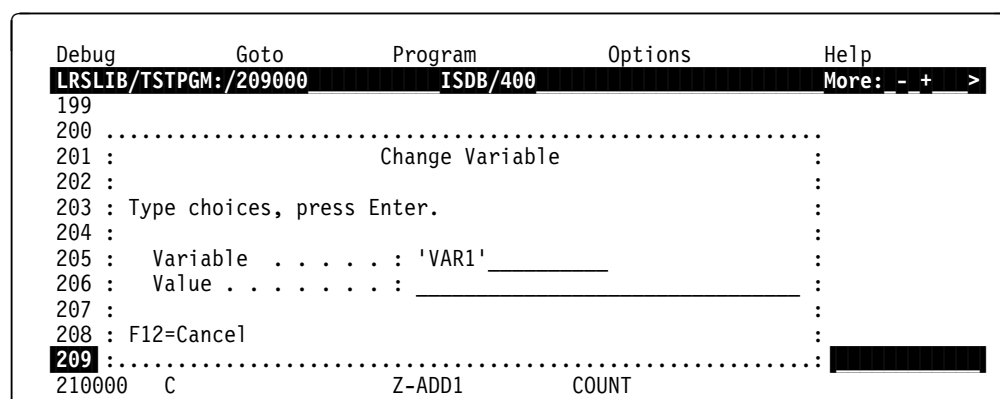


Figure 8. Change Variable Window

3. In the *Value* prompt, type the value you want to assign the variable and press Enter. You can specify a hexadecimal value in the prompt if you want.

To change the value of a variable that is not visible on the Source display:

1. On the command line, type CHG, followed by the variable's name, and then its new value. You can specify a hexadecimal value if you want.
2. Press Enter. The Source display appears, and the value of the variable is changed.

Examples

```
CHG var1 '  Test'
```

The first eight characters in the character string contained by the variable `var1` are changed to ' Test'.

```
CHG (60)address 'xxxxxxxx'
```

The substring from characters 60 to 68 in the contents of the variable `address` is replaced with 'xxxxxxxx'.

```
CHG var1 999
```

The value of the variable `var1` is changed to the numeric value 999.

```
CHG ARR(2) 40
```

The value of the second element of the RPG array `ARR` is changed to the numeric value 40.

```
CHG var1 X'00058F'
```

The value of variable `var1` is changed to the hexadecimal value X'00058F'.

Watching Variables

You can monitor, or watch, up to 15 program variables in the Source display.

To monitor variables:

1. Type the `WATCH` command, followed by the name of the variable you want to monitor. To monitor its hexadecimal value, use the `WATCHHEX` command instead.
2. Press Enter. The variable, if it can be found, is displayed with its current value under the status line.

A message tells you if the variable cannot be found. If the variable is found but has no current value, a message is displayed and the variable is shown with a value of an asterisk (*) and a message number.

```

Debug          Goto          Program          Options          Help
LRSLIB/TSTPGM:/209000          ISDB/400          More: _ - + _ >
COUNT= '10'
TOTAL= '35'
COUNT=X'0000010F'
202000 C          Z-ADDAMT          TEMP
203000 C          TEMP          MULT 100          TEMP00
204000 C          TOTAL          ADD TEMP00          TOTAL
205000 C          WRITESUBFILE
206000 C          ADD 1          COUNT
207000 C          COUNT          IFGT 100
208000 C          Z-ADD0          TEMP00
209000 C          Z-ADD0          TOTAL
210000 C          Z-ADD1          COUNT
211000 C          EXCPTOUT
212000 C          END
213000 C          READ FILE          99
214000 C          FLAG          COMP '*'          23
215000 C          *IN23          IFEQ '1'
216000 C          MOVE 'DONE'          TEMP01
ISDB ==>
F3=Exit      F5=Step  F6=Break  F11=Display variable
F12=Cancel  F17=Run   F23=Change variable  F24=More keys
Stopped at statement 209000.

```

Figure 9. Source Display with Watched Variables

Examples

WATCH count

The value of the variable *count* will be displayed in the Source display.

W ARR(2)

The value of the second element of the RPG array *ARR* will be displayed in the Source display.

WH temp00

The hexadecimal value of the variable *temp00* will be displayed in the Source display.

Removing Watched Variables from the Source Display

You may want to stop watching variables once you are finished monitoring them, since the addition of each watched variable reduces the amount of source you can view in the Source display.

To stop watching a variable and remove its entry from the Source display:

1. Type the UNWATCH command, followed by the name of the variable you do not want to watch.
2. Press Enter. The variable and its value are removed from the Source display.

To stop watching all variables, type the UNWATCH * or UNWATCH *ALL commands.

Displaying Information about a Program

The Source display provides access to information that is relevant to the job in debug mode and the program you are debugging.

The Program pull-down window, for example, provides access to AS/400 commands that you may need.

Debug	Goto	Program	Options	Help
LRSLIB/TSTPGM: /209000		ISDB/400		More: - + >
199000	C	PRCSFL	1. Display object description	
200000	C		2. Display program	
201000	C	*IN99	3. Display program reference	
202000	C		4. Work with job	
203000	C	TEMP		
204000	C	TOTAL		
205000	C			

Figure 10. Program Pull-Down Window

You can use this pull-down window to do the following:

- To view information about the program you are debugging in the AS/400 Display Object Description display, select option 1 (Display object description). This option runs the AS/400 DSPOBJD command.
- To view information about the attributes of the program you are debugging in the AS/400 Display Program Information display, select option 2 (Display program). This option runs the AS/400 DSPPGM command.
- To view information on the programs called and file objects being used, select option 3 (Display program reference). This option calls the AS/400 DSPPGMREF command and displays its output in the Display Spooled File display.
- To work with the job being debugged in the AS/400 Work with Job display, select option 4 (Work with job). This option runs the AS/400 WRKJOB command.

In addition to using the Program pull-down window, you can select Option 3 (Display debug) in the Debug pull-down window to view information about the program stack and the names of programs in debug mode.

Working with More than One Program

You can work with up to ten related programs in the interactive source debugger environment. These programs are kept in a program list. You can view this list at any time in the Work with Program List display. You can use this display to:

- Add a program to the debugging environment, so that you can debug it if it is called by the program you are debugging.
- Remove a program from the debugging environment.
- Select which program you want to view in the Source display so that you can set breakpoints.

To view this display, select option 5 (Work with programs) from the *Debug* pull-down window. Alternatively, press F14 from the Source display.

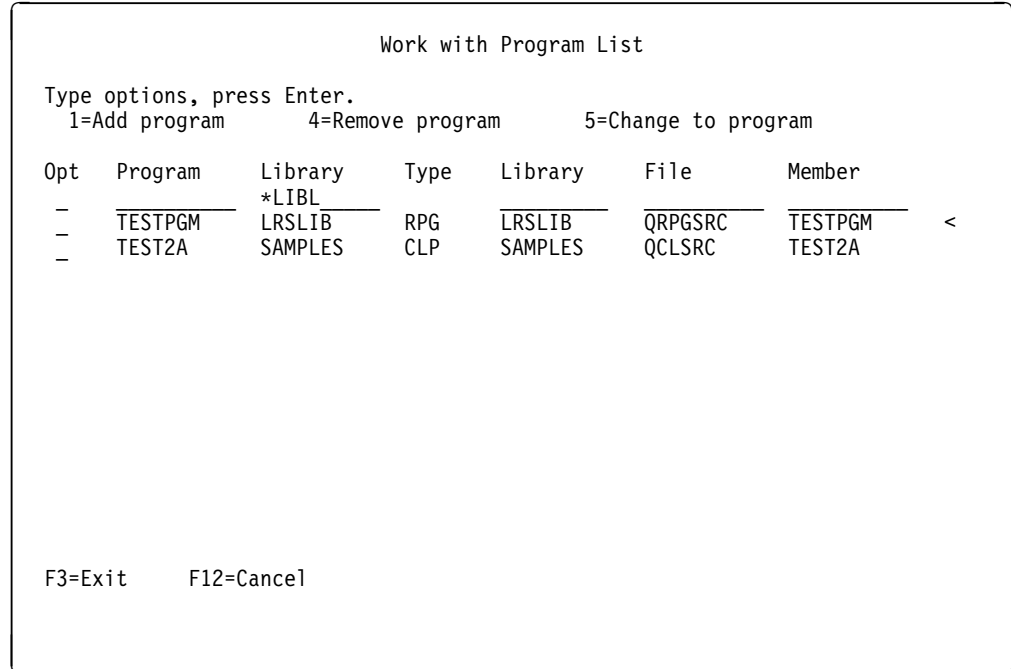


Figure 11. Work with Program List Display

Adding a Program to the Debugging Environment

To add a program to the interactive source debugger environment so that you can debug it if it is called by the program you are debugging:

1. Press F14 from the Source display. The Work with Program List display appears.
2. Select option 1 (Add program) in the *Opt* column for the first line. This row has a blank entry in the *Program* column.
3. Type the name of the program under the *Program* heading.
4. If the program is not in your library list, type the name of the library that contains it over the *LIBL value in the *Library* column.
 (Use the value *CURLIB if you want to use the current library.)
5. If you want source appearing in the Source display different from that associated with the program, specify the names of its library, file, and member in the fields provided.
 (If the sequence numbers in the specified source do not match those of the program being debugged, the results may be unpredictable.)
6. Press Enter.

Alternatively, use the ADDISDBPGM command. Refer to the online help for more details on how to use this command.

Changing the Program You Are Viewing

Once you have added a program to the debugging environment, you can view its source and set breakpoints in it to customize how you will debug it when it is called by another program.

To view the source of a different program in the Source display:

1. Press F14 from the Source display. The Work with Program List display appears.
2. Select option 5 (Change to program) in the *Opt* column beside the program you want to work with.
3. Press Enter. The source of the new program appears in the Source display so that you can set breakpoints. The program you were debugging remains in debug mode, but you cannot work with it until you select it again using the Work with Program List display.

Alternatively, use the CHGISDBPGM command. Refer to the online help for details.

Removing a Program from the Program List

Since the debugging environment cannot support more than ten programs at a time, you might need to remove one program so that you can work with another. To do this:

1. Press F14 from the Source display. The Work with Program List display appears.
2. Select option 4 (Remove program) in the *Opt* column beside the program you want to remove.
3. Press Enter. The program you selected is removed from the debugging environment.

Alternatively, use the RMVISDBPGM command. Refer to the online help for details.

Debugging Programs in Other Jobs

To debug a program running in another user job:

1. Type STRISDB on the AS/400 command line, and press F4.
2. Type the name of the program you want to debug. (For more information, see “Starting the Interactive Source Debugger” on page 2.)
3. In the *Job to service (SRVJOB)* prompt, type *SELECT.
4. Press Enter. A list of active jobs on the system appears.
5. Select option 1 for the job you want to put in debug mode, and press Enter. The interactive source debugger tool will issue a STRSRVJOB command for that job, and go into single step mode. The Source display for the program being debugged will appear.

(To debug a batch job, it must be running and not held on the job queue.)

Working with Source on Another AS/400 Machine

It is possible to debug a program even when the program source is on a different machine. To do this:

1. Use the CRTDDMF command to create a DDM file that points to the source member you want to use.
2. Type STRISDB, and press F4.
3. Type the name of the program you want to debug.
4. If the program was created using a DDM file, ISDB searches for its source on the other machine. Otherwise, specify the names of the library, file, and member of the DDM source member you want to use in the fields provided.
5. Press Enter to start ISDB.
6. When you call the program and the Source display appears, the source it contains will be the source from the other machine.

Bibliography

AS/400

AS/400 Programming: Command Reference Summary, SC21-8076.

AS/400 Programming: System Reference Summary, SC21-8104.

AS/400 System Operation: Operator's Guide, SC21-8082.

Publications Reference, SC41-3003.

IBM CL

AS/400 Control Language Reference, SC41-3722.

AS/400 Programming: Control Language Programming, SC41-3721.

IBM RPG/400

AS/400 Languages: RPG/400 Reference, SC09-1817.

RPG/400 Reference Summary, SC09-1525

AS/400 Languages: RPG/400 User's Guide, SC09-1816.

IBM Licensed Program Specifications for RPG/400, GC09-1828.

IBM Common Programming Interface: RPG Reference, SC09-1286.

IBM COBOL/400

AS/400 Languages: COBOL/400 Reference, SC09-1813.

AS/400 Languages: COBOL/400 Reference Summary, SX09-1286.

AS/400 Languages: COBOL/400 User's Guide, SC09-1812.

Application Development ToolSet/400

- *ADTS/400: Advanced Printer Function*, SC09-1766
- *ADTS/400: Character Generator Utility*, SC09-1769
- *ADTS/400: Data File Utility*, SC09-1773
- *ADTS/400: File Compare and Merge Utility*, SC09-1772
- *ADTS/400: Interactive Source Debugger*, GC09-1897
- *ADTS/400: Programming Development Manager*, SC09-1771
- *ADTS/400: Report Layout Utility*, SC09-1767
- *ADTS/400: Screen Design Aid*, SC09-1768
- *ADTS/400: Screen Design Aid for the System/36 Environment*, SC09-1893
- *Introducing Application Development ToolSet/400 and the AS/400 Server Access Programs*, SC09-1939

The orderable features of 5763-PW1 are:

- *ADTS/400: Application Development Manager Introduction and Planning Guide*, GC09-1807
- *ADTS/400: Application Development Manager User's Guide*, SC09-1808
- *ADTS/400: Application Dictionary Services Self-Study Guide*, SC09-1904
- *ADTS/400: Application Dictionary Services User's Guide*, GC09-1860

Index

A

- adding a program to the debugging environment 31
- ADDISDBPGM command 31
- alias
 - See equate name
- array elements for RPG, displaying 25
- arrows indicating RPG program flow, setting 19
- AS/400 commands
 - ADDBKP, restriction 12
 - DSPOBJD 30
 - DSPPGM 30
 - DSPPGMREF 30
 - ENDISDB 6
 - entering from the Source display 10
 - STRISDB 2
 - STRISDB, working with source on another AS/400 machine 33
 - STRSRVJOB 32
 - WRKJOB 30
- AS/400 displays
 - Display Debug Information 23
 - Display Object Description 30
 - Display Program Information 30
 - Display Program Variables 25
 - Display Spooled File 30
 - Work with Job 30
- attributes of a program, viewing 30

B

- BACKWARD command 15
- batch job, debugging 32
- BOTTOM command 15
- BREAK command 21
- breakpoints 19
 - definition 21
 - removing 23
 - running through a program until reached 23
 - setting 21
 - setting conditional 22
 - setting how highlighted 19
 - setting if statement run several times 22
 - viewing list of 23

C

- calling other programs 30
- case-sensitive search, specifying 17
- Change Variable window 27
- changes from Version 2 ix
- CHG command 27

- CHGISDBPGM command 32
- CL variable name, use of in commands 12
- CLEAR command 23
- clearing equate names 13
- colors in Source display 18
- Command Entry window 10
- command keys 9
- command line, definition 8, 10
- commands
 - See also AS/400 commands
 - ADDISDBPGM 31
 - assigning shorter name to 12
 - BACKWARD 15
 - BOTTOM 15
 - BREAK 21
 - CHG 27
 - CHGISDBPGM 32
 - CLEAR 23
 - DOWN 15
 - DSP (display variable) 25
 - DSPHEX 25
 - EQUATE 12
 - FIND 16
 - for invoking pull-down windows 8
 - FORWARD 15
 - GOTO 15, 16
 - LABEL 16
 - LEFT 15
 - logging 14
 - notes on using 12
 - PRT 26
 - QUIT 6
 - RESTART 6
 - retrieving 10
 - RFIND 16
 - RIGHT 15
 - RMVISDBPGM 32
 - SCREEN 8
 - SET BREAK 19
 - SET COLOR 18
 - SET CURSOR 19
 - SET FLOW 19
 - SET LOG 14
 - SET ROLL 19
 - short forms for 12
 - STEP 24
 - summary 10
 - SYS 10
 - TOP 15
 - UNWATCH 28
 - UP 15
 - USE, for equate names 13

commands (*continued*)
 USE, to use command log again 14
 WATCH 28
conditional breakpoint, setting 22
cursor behavior 19

D

date stamp, including in text search 17
Debug menu-bar choice, summary 9
debugging environment, ten-program restriction 30
direction of search, specifying 17
Display Debug Information display, viewing 23
Display Program Variables display 25
display, viewing previous 8
DOWN command 15
DSP (display variable) command 25
DSPHEX command 25

E

editing source, restriction 1
elements in RPG arrays, displaying 25
ending ISDB 6
ENDISDB command 6
EQUATE command 12
equate name 12
error in program producing unmonitored message 24
exact match, specifying for text search 17
executing a program 23
extended help 6
externally described files, restriction 1

F

FIND command 16
Find String window 17
finding a place in a program 15
finding text 16
FORWARD command 15
function key sets, viewing 8
function keys 9
future session, saving settings for 14

G

GOTO command 15
Goto menu-bar choice, summary 9
Goto pull-down window 15
Goto Subroutine window 15

H

Help menu-bar choice, summary 9
help, using 6
hexadecimal values of variables
 changing 27

hexadecimal values of variables (*continued*)
 displaying 25
 monitoring 28
 printing 26
highlighting in Source display 18

I

indicators for RPG, displaying 26
information on a program, displaying 30

J

jobs
 debugging in other 32
 serviced 3, 6
 working with 30

K

keys
 summary 9

L

LABEL command 16
LEFT command 15
line in program, viewing 15
listing views, restriction 1
logging commands and actions 14

M

menu bar, definition 7
menu-bar choices, summary 9
message line, containing unmonitored message 24
message line, definition 8

O

online help, using 6
Options menu-bar choice, summary 9
other programs, debugging 30

P

prerequisites for using ISDB vii
printing program variables 26
profile, using 14
Program menu-bar choice, summary 9
Program pull-down window 30
PRT command 26
pull-down window, using choices in 8

Q

QUIT command 6

R

removing a program from the debugging environment 32
removing breakpoints 23
reproducing actions in another session 14
RESTART command 6
restrictions
 /COPY files 24
 AS/400 debug commands 12
 batch jobs 32
 editing source 1
 equate names 12
 externally described files 1, 24
 labels, maximum number 16
 listing views 1
 ten related programs 30
retrieving commands 10
RFIND command 16
RIGHT command 15
RMVBKP AS/400 command, restriction 12
RMVISDBPGM command 32
RPG
 array elements, displaying 25
 indicators, displaying 26
 subroutine, viewing 15
RPG Indicators 1 - 99, L1 - L9 window 26
running a program 23

S

SCREEN command 8
scrolling through a program 15
scrolling values in Source display, changing 18
searching for text 16
serviced jobs 3, 6
SET BREAK command 19
SET COLOR command 18
SET CURSOR command 19
Set Display Options window 18
SET FLOW command 19
SET LOG command 14
SET ROLL command 19
Source display
 definition 7
 displaying 5
 setting arrows indicating program flow 19
 setting colors 18
 setting cursor behavior 19
 setting highlighting 18
 setting how breakpoints are highlighted 19
 setting options 18
 setting scrolling and shifting values 18
 summary of commands 10

Source display (*continued*)
 summary of function keys 9
 summary of menu-bar choices 9
source, using from another AS/400 machine 33
statement currently executing, viewing 15
statement in program, viewing 15
status line, definition 7
stepping through a program 24
string of text, finding 16
STRISDB command
 examples of 5
 using to debug another job 32
 working with source on another AS/400 machine 33
substrings of variables 24
summaries
 changes from Version 2 ix
 commands 10
 function keys 9
 menu-bar choices 9
SYS command 10

T

text string, finding 16
TOP command 15

U

unmonitored AS/400 messages 24
UNWATCH command 28
UP command 15
update production value, changing 3
USE command
 for equate names 13
 for using command log again 14
user job, debugging another 32

V

variables
 changing 27
 monitoring 28
 printing 26
 substrings 24
 viewing 25

W

WATCH command 28
watching program variables 28
Work with Program List display 30