

XRAY is a cost effective interactive source-level debugging system for batch programs. It enables fast and effective development, testing, and maintenance of batch applications, improving programmer productivity and encouraging the production of more resilient programs to improve the reliability of production systems.

XRAY's features include:

- A menu based command structure
- Point and click techniques for navigation and display
- Source level debugging with automatic display of the variables referenced by a the current statement
- Abend interception
- Wild branch detection
- Program loop detection
- Conditional and unconditional user defined halt points
- Display and modify data in storage
- Define items to be displayed on demand or at each program stop point
- Debug statically linked or dynamically called subprograms
- Alter program execution path (skip around or intentionally test specific code, step through program statements)
- Audit trail to log details of changes made during debugging
- Comprehensive help facilities



XRAY Benefits

- Improved programmer productivity from debug sessions by permitting the resolution of multiple problems in a single session
- Reduced development times due to improved resolution of problems
- Better tested and more resilient programs
- Ability to test infrequently used logic paths to identify and correct obscure program errors
- Improved program maintenance
- Reduced dump analysis
- Improved programmer morale
- Reduced overall cost from significantly improved productivity and fewer production outages due to reduced error levels

XRAY Features in Detail

Ease of Use

The XRAY system is easy to use being menu and PF key driven system with on-line help screens. The interactive screens that highlight error conditions are easy to understand making the system an outstanding learning aid for junior programmers who can easily follow program logic and get around the problems always encountered by new programmers.

Point and Click

Wherever feasible commands are effected by positioning the cursor and then pressing an appropriate function key to cause a command to be executed. This technique is used on all menus to select a function, on source displays to select variables for display, on source and link data directories to select source or link data for display and on other displays to select items for expansion or deletion.

Menu Based Command Structure

All XRAY facilities are accessible via clearly informative menu screens. Each menu provides an indication of the functions available to allow easy use of them. Options may be selected from a menu by positioning the cursor on the appropriate line and pressing the ZOOM PF key. Alternatively the option number shown on the left of the option description may be typed into the command field that is present on every XRAY display.

Each line on the menu also shows the fast path id associated with the option. These fast path ids can be used by more experienced users to navigate directly to any XRAY display from any point in the system.

Program Monitoring

Provides abend interception, wild branch detection and loop detection. Any program check or abend request in a monitored program results in a stop display at the point of the abend. Similarly any attempt by the program to branch to a random point results in a stop display showing details of the wild branch.

Limits may be set on the number of instructions that may be executed between program halts. If the limit is exceeded, an automatic halt occurs. This allows the detection of program loops even when they involve I/O requests..

Source Level Debugging

The programmer's source code is captured at compile time and stored on the XRAY source file. It can then be displayed on-line and used for the setting of halt points during a debugging session. Halt points are set and reset by simply typing a one character command on the statement number of the displayed source.

When a stop is encountered, the source at that point is displayed together with the value of variables referenced in the current statement. These values can be altered by simply typing the new value.

For COBOL programs an expanded view of data structures can be displayed using the GROUP command. This gives a fully formatted display including all data types and subscripted areas.

Control Debugging and Execution using line commands

A debug session is made easy using LINE commands. You can control where execution of a program is halted, specify special halt conditions, and even alter the logic flow of the program using single character LINE commands. All of this can be done while viewing the source code of the program.

Unconditional Halt Points

With XRAY program execution may be suspended at programmer-defined halt points.

Unconditional halts always cause the program to be stopped and are easily set by typing a one character command on the appropriate program statement number when the source is displayed. They can also be set using an Add Halt Point display where certain specialized halt offsets such as START can be used.

When a halt point is reached a stop for a 'halt request' is presented giving details of the current statement and variables referred to by the statement

When any halt occurs, all XRAY debugging facilities are available for the examination and alteration of data in storage and of program code. This enables errors to be detected and corrected or changes made to ensure execution of particular sections of code.

Conditional Halt Points

Conditional halts are also easily set by typing a one character command on the appropriate program statement. For a conditional halt however, a secondary screen is displayed to enable the condition to be provided. This is normally in the form of a comparison but may also include limits to the number of times a halt will occur and how often the halt will occur.

Conditional halts can also be set that are independent of any program statement using the Add Halt Point display. This can be especially useful to detect where a variable is changed or is set to a particular value.

Define Items for Display on Demand

Storage areas or program variables may be selected to be displayed at each program stop by defining them as Keep items. This is easily achieved by positioning the cursor on the name of a variable shown on a source display and pressing the appropriate function key. Once defined a keep item may be displayed in a data window on the Stop Display or in response to the K (keep) command. The items may be displayed either by value (i.e. in accordance with the data definition for the item) or in hex. There is no limit to the number of items that may be defined in this way.

Display and Modify Data in Storage

Storage areas or program variables may be displayed and altered singly or collectively in a number of ways. The display may occur as the result of a direct display command, as the result of a variable being acted upon by the current program statement, as the result of a Group variable display command or a Keep item display command. However, the data display originates, the data displayed may be changed simply by over-typing the existing value. If data is displayed in hex and character format, changes can be made to either form of the display. This makes it possible to assign incorrect values to areas to help test data analysis logic.

Debug Subprograms

In the situation where multiple components of a program are to be debugged simultaneously, XRAY permits halt points to be established in any component of the program whether statically linked to the main program or called dynamically. For statically linked components it is important that the link map for the program is

stored on the source file to enable the names and locations of each component to be established. For dynamically called programs, halt points can be established before the program is loaded and will be held as inactive by XRAY until the program is loaded at which point they will become active and cause a stop when a halt point is reached.

Alter Program Execution Path

At a program stop for whatever reason execution can be continued at a point other than that at which the program was last halted. This allows the bypassing of erroneous code, the simulation of conditions which are hard to create and the testing of infrequently used logic paths.

Single-Stepping

At a halt, program execution may be continued by instruction stepping with a further halt after each step. A step may be one or more program statements or machine instructions. It is also possible to step immediately to a specified location in a halted program. Stepping is normally confined to the current program module to avoid stepping through compiler generated logic needed to handle complex statements. However, the scope of the stepping operation can be widened to encompass the whole program or multiple programs.

Program Flow Analysis

XRAY maintains a table of the most recent branches taken during program execution. At any stop point the flow command can be used to display this information to show the program's recent execution path and from that it is possible to determine why the program reached the stop point and whether it has reached that point correctly or as the result of a logic error. The display identifies the 'from' and 'to' statement numbers when source data is available and the names of called modules when link edit data is available.

Modification Log

Any changes made to storage areas when using XRAY are logged to a CICS transient data destination. The log identifies the individual that made the change and identifies the time at which the change was made and the terminal used to make the change.

XRAY Benefits in Detail

Faster Debugging

XRAY provides faster debugging of program logic, coding, and data errors. It provides for user halts to be defined at the statement level either in the main program or in any called subroutine. Program execution can also be halted at programmer defined halt points that are activated only when certain conditions are encountered. Data can be displayed symbolically using variable names. Loops can be trapped by setting an instruction limit.

Faster Corrections

XRAY pinpoints errors for you on the screen. COBOL, PL/1 and Assembler source can then be displayed to aid the debugging process. Corrections to variables and data are made interactively and execution of the program then continues. Program flow after a halt can be redirected to test infrequently used logic paths.

Faster Turn Around

Multiple errors can be examined and resolved in one execution of the program reducing the number of times a program has to be recompiled to enable further testing.

Program Abend Interception

When a monitored program abends, XRAY intercepts the error and displays the associated program source code, the reason for the abend, and other related information. All XRAY debugging facilities are available at this point for problem determination. Once the fault has been identified the programmer can decide what course of action to take. For example, the programmer could correct the problem and continue execution or bypass the code in error entirely.

Improved Program Maintenance

Problems in even the most unfamiliar of existing systems can be pin-pointed and corrected. Everything needed is available on-line. Inherited code which needs to be understood before modification may be investigated first and tested afterwards using, for example, features such as flow analysis, instruction stepping and storage display.

Reduced Dump Analysis

It is no longer necessary to produce core dumps for each program error. XRAY displays exactly what you need to know on the screen at the time error occurs. It takes the drudgery out of debugging and testing.

Supported Environments

Systems: VSE/ESA, z/VSE: all releases
OS/390, z/OS: all releases
CICS: 1.7 - TS 1.1 (VSE) or TS 3.1 (z/OS)

Languages: Cobol, Cobol II, COBOL for MVS/VSE, Enterprise COBOL, PL/1 for MVS/VSE, Enterprise PL/1, and Assembler

Pricing

XRAY is competitively priced. Most equivalent products are considerably more expensive. XRAY upgrade charges are also minimal whereas other products often result in large upgrade fees when upgrading your operating system release or CPU. Sometimes, these fees apply even if the capacities of the old and new CPUs are the same!

Free Trial

See the benefits yourself with a no obligation free trial as XRAY is easy to install and requires no modifications to the operating system.

Distributor for Australia and New Zealand:

Generic Systems Australia
Level 21, 201 Miller St.
North Sydney NSW 2060

Tel.: 61-2-9959-2239
Fax: 61-2-9997-5411
Email: mktg@gensys.com.au

