

HP Visual Inspect Software

Data sheet



HP Visual Inspect Software offers an open debugging architecture that is flexible, easy to use, and capable of addressing distributed, heterogeneous applications. Visual Inspect Software provides debugging support for explicitly parallel instruction computing (EPIC), RISC, and CISC machine architectures and compilers, and the HP Guardian and Open System Services (OSS) environments.

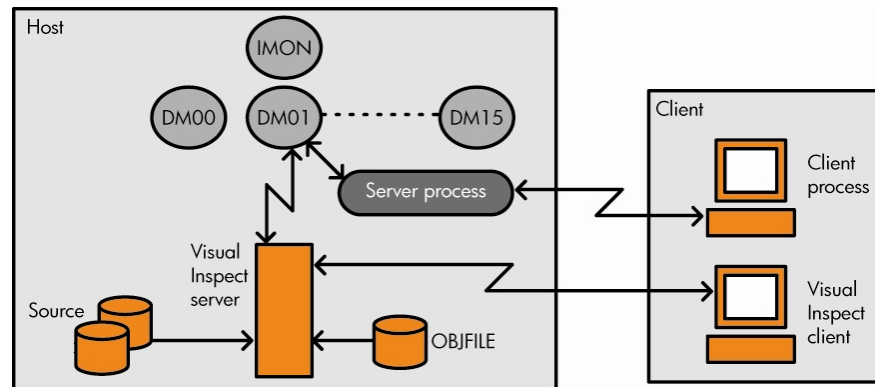
Users can debug multiprocess applications, where processes may exist on a single machine or may be distributed across a network. They are also able to perform post-mortem debugging, where application-state information is saved for later analysis.

Key features and benefits

- Powerful and easy-to-use interface
- System- and application-level debugging
- Support for distributed, multiprocess applications
- Program transfer support
- Easy program navigation using various views and graphical user interfaces (GUIs)
- Support for post-mortem debugging

Visual Inspect Software is a Microsoft® Windows® system-based debugger that is flexible, easy to use, and capable of debugging distributed, heterogeneous Integrity NonStop server applications. Debugging support is available for EPIC, RISC, as well as CISC applications executing in the Guardian and OSS environments.

Figure 1
Visual Inspect architecture



The open debugging architecture of Visual Inspect Software is shown in figure 1.

Visual Inspect Software provides a Win32 client with a GUI. This interface uses program visualization, direct manipulation, and other techniques to significantly improve debugging efficiency. The visual interface simplifies program navigation, comprehension, and multiprocess debugging.

Powerful and easy-to-use visual interface

Visual Inspect Software provides support for debugging small to large programs and multiprocess applications. The interface uses techniques to improve user productivity, such as persistent debugging options, and has dynamic display and the ability to set multiple watch windows. The GUI has been designed to be both intuitive to new users and powerful for sophisticated ones.

The Visual Inspect debugger allows application execution to be viewed and controlled at a variety of abstraction levels, including the following:

- *Application level*—At this level the user can see and manipulate such application elements as processes, open data files, and communications connections between processes. A user can activate or suspend all of the processes that compose the application.
- *Process level*—The user can see and manipulate the resources that are associated with a particular program, such as open files, allocated data segments, and program source files. This level provides various mechanisms for navigating: displaying and manipulating scope static call hierarchy and the modules that compose the program.
- *Source level*—At this level the user can see and manipulate elements of the program source text. Source text is annotated to provide additional information about the lexical structure of the program and about compiler optimizations. Users can display structures and modify the values of program variables and set breakpoints.

System- and application-level debugging

System software developers are sophisticated users of debugging tools and are knowledgeable about system internals and machine architecture. They debug at both the source and machine levels and may need to examine system data structures. Visual Inspect Software aids system-level debugging by providing support for user processes and processes executing on standalone systems. The program transfer into and from Visual Inspect Software enables the system programmer to gain access easily to native command-line Inspect Software features for lower-level debugging.

Application programmers debug user processes; in the case of multiprocess applications, they may need to debug several such processes at the same time. Application developers tend to debug at the source level and usually do not need to examine the machine state. Visual Inspect Software provides application programmers with a source-level debugging model, support for multiprocess and application-level debugging, improved expression evaluation, and graphical interfaces to make invoking functions easy.

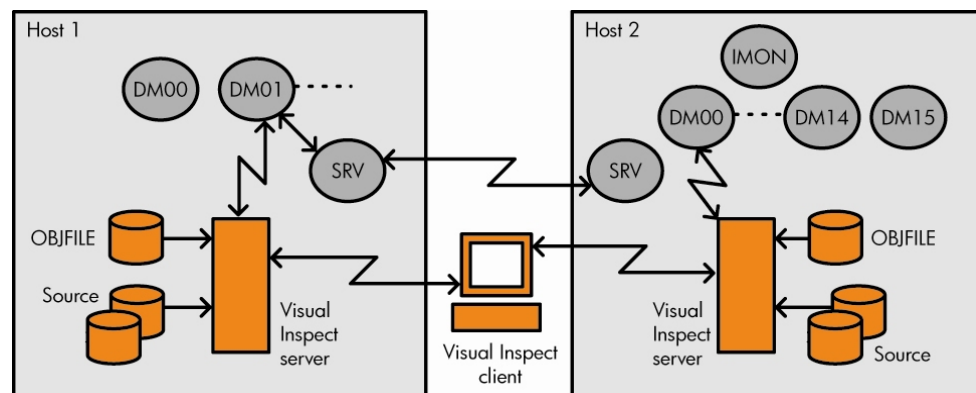
Visual Inspect Software also addresses the needs of the production environment with its ability to

- Debug processes executing object files stripped of symbols, using symbols from an alternate object file
- Specify source file locations when they have changed from their original compiled location
- Attach to and detach from running processes

Support for distributed, multiprocess applications

The Visual Inspect client/server architecture supports the efficient debugging of distributed applications. A single client may communicate with servers on multiple machines to debug processes running there (see figure 2). The client gives you a single, unified view of the application, while having a server on each machine improves debugging performance.

Figure 2
Distributed system debugging



Program transfer support

Visual Inspect Software provides program transfer support for moving a program to and from Inspect Software during a debugging session. It is useful for using Visual Inspect Software to debug programs that have entered the control of Inspect Software. It is also useful to access the additional functionality currently available in Inspect Software.

Easy program navigation using various views and GUIs

Visual Inspect Software provides views of applications at various levels so programmers can see their programs in multiple ways. Visualization levels include Scope Browser, Application Control View, Call Hierarchy View, Program Control View, Resource Manager, Watch Window, Scope Source, and Find Scope. With these visualization interfaces, the developer can navigate from high to low level and can display and/or perform the following:

- A representation of the processes that compose an application and indicate their execution states
- Any process that composes an application
- Open file and execution environment information
- Windows associated with a particular process so they can be identified and managed
- Online help for external commands and capabilities, context-sensitive help for specific features, and diagnostic and error messages
- Display and navigate static call graph
- View scopes as organized by compilation unit

Scope Browser

The Scope Browser lets you

- View the hierarchy of program units (linkage units, modules, and scopes) in your program
- Get detailed information on a selected scope, module, or linkage unit using the Properties button
- Set and clear breakpoints at the entry points of scopes
- Open a Scope Source view to examine source
- Open a Call Hierarchy view beginning at a selected scope

Application Control View

Application Control View lets you

- View a list of programs currently being debugged in your Visual Inspect session. The list includes the following information for each program:
 - Program: The name of the program file
 - Type: An executable or snapshot file
 - State: The current state (Run, Stop, or Hold)
 - System: Where the process is running (or, if a snapshot file, where it was created)
 - User: The user ID of the service connection
- Get detailed information on a selected program using the Properties button.
- Display an expression in a Watch Window. The expression will be evaluated in the context of the selected program.

- Open another view to get more information about a selected program.

Call Hierarchy View

Call Hierarchy View lets you

- View the static call tree of your program
- Navigate to scopes that are not on the stack
- Get detailed information on a selected scope using the Properties button
- Set and clear breakpoints at the entry points of scopes
- Open a Scope Source view to examine source

Program Control View

Program Control View lets you

- View the lines of source and browse through them, navigating among active scopes by selecting stacks from the drop-down list
- Obtain information about the source properties using the context menu
- Control program execution
- Set breakpoints on individual statements
- Display and modify locals, parameters, variables, and expressions, and browse arrays
- Copy and paste selected source text into dialog edit boxes or other Windows applications
- Identify whether there is a timestamp mismatch within the source
- Set a source assign

Watch Window

Watch Window lets you

- View the name, value, scope, type, and location of a display item (a variable, expression, or array element)
- Monitor values of display items over time as the program executes, and identify changed values (red) and out-of-scope data (gray) by color
- Add and delete items to display in the Watch Window
- Modify the value of a displayed item and have the change evaluated in the context of the current scope
- Change the formatting, style, or type of a displayed item
- Freeze and thaw window contents
- Select and rearrange fields displayed in the window
- Sort items by name, scope, or location

Scope Source

Scope Source lets you

- View the source for the selected scope.
- Display static memory entities such as constants, macros, and literals. Because there is no stack frame associated with the Scope Source view, only program entities that do not occupy dynamic memory can be displayed. Entities occupying dynamic memory are only accessible through the Program Control View.
- Place breakpoints using statement marker.

- Resume, Hold, or Stop the program.

Find Scope

Find Scope lets you

- Search for a scope with a name matching the specified name or pattern. You can restrict the search using linkage unit, module, and scope criteria. Pattern matching displays a list of scopes from which to choose.
- Get detailed information on a selected scope using the Properties button.
- Open a selected scope in Scope Source view to examine source.
- Open a Call Hierarchy starting at the specified scope.

Resource managers

Resource managers let you monitor (and in some cases change the properties of) resources being used by Visual Inspect Software or the program you are debugging:

- *Breakpoints*: View, set, alter, and clear Code, Data and Event breakpoints by using the toolbar buttons on the manager. For Code and Data breakpoints you can also reset the hit counter to zero and set other breakpoint options.
- *Connections*: View, delete, and create connections to Integrity NonStop server hosts, and view connection and host properties.
- *Linkage units*: View the names and symbol files of the linkage units (executable or library object files) that compose the program you are debugging, and view linkage unit properties and change the symbol file associated with a linkage unit.
- *Opens*: View the names of the files that the program you are debugging has opened and their properties.
- *Segments*: View information about the extended segments allocated by the program you are debugging.
- *Signals*: View a list of signals, handlers, masks, and flags for the program you are debugging and their properties.
- *Source assigns*: View the Original and Current paths of source files and redefine the location of source files.
- *Source search path*: View the list of subvolumes and directories in which Visual Inspect Software looks for source when it is not found at its presumed location. You can rearrange the order in which Visual Inspect Software searches the paths.
- *Stack frames*: View the scope name, line, and statement number of each stack frame and view frame properties, display variables with a frame context, and view source and the call hierarchy for a selected frame.

Persistence

Using persistence, Visual Inspect Software preserves certain user settings and values and then recreates them automatically. There are two types of persistence:

- *Session persistence* includes the default settings that you applied in the previous session, regardless of the program you were debugging. The next time you start the client, the fields displayed and the order of the fields are the same as you left them at the close of the previous session. Session persistence includes
 - Size and position settings for all managers and windows except Watch Window
 - Column sizes for Watch Window
 - Location of the floating toolbar in the Application Control View
 - Configured view on the Breakpoints Manager and Watch Window

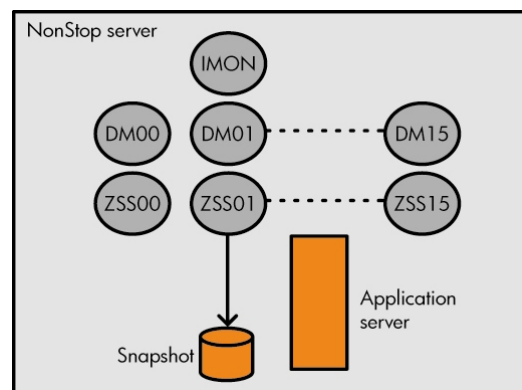
- *Program persistence* includes the settings that you applied when debugging a specific program. Program persistence includes
 - Code and Event breakpoints set in the program
 - Size and column widths for Watch Window and Program Control View
 - Configured view on the Watch Window
 - The relative proportions of the source pane and Locals/Parameters panes, locals sort type, and whether or not the Locals/Parameters pane is visible

Program persistence overlaps and overrides session persistence.

Support for post-mortem debugging

Visual Inspect Software has a component called the snapshot server, which handles snapshot (save) file creation. One snapshot server executes in each processor. The mechanism addresses capacity limits that previously existed with the Inspect subsystem. This improvement enables new mechanisms to be developed for creating snapshot files, such as a procedural interface that a process could use to save its own state. Figure 3 depicts the role of the process snapshot server in Visual Inspect Software.

Figure 3
Process snapshot



New features

The following are the new features added in the Visual Inspect Version 3 Software. Visual Inspect 3.0 adds support for debugging applications running on Integrity NonStop servers as well as many new debugging features on the servers, including low-level debugging capabilities. With Visual Inspect 3.0, users can

- Debug EPIC and CISC application programs running on Integrity NonStop servers
- Debug snapshot files from EPIC, RISC and CISC applications
- Continue to debug RISC and CISC application programs running on HP NonStop S-series servers (running Visual Inspect Server Version 2.4)

And applications running on Integrity NonStop servers (running the Visual Inspect Server Version 3.0)

- Display new data types
 - Unsigned long long and _ptr64 for C/C++
 - EXT64ADDR and .EXT64 for pTAL
- Display arbitrary memory; value can be formatted, modified, and monitored
- Can create a watch item from an address; the watch item can be formatted as user defined type, system defined type, or language type
- Display assembly language instructions interleaved with associated source code statements
- Step instructions
- Selectively read dynamic link library (DLL) files, effectively controlling the performance of symbol searches and facilitating the debugging of applications with large number of DLLs
- Display watch items, source code comments, and other non-compiled text using multiple byte character sets
- Set conditional memory access breakpoint
- Default the data range for the memory access breakpoints to the size of the symbolic data item being monitored, or to the length of the string for conditional memory access breakpoints that compare strings
- More easily debug privileged code
- Allow the super user to enable privilege access to
 - Open program now provides an option to hold a process even if it is in privilege code
 - Display and format Subsystem Programmatic Interface (SPI) and Event Management Service (EMS) databuffers
 - Switch to the system debugger for command-line debugging and back to Visual Inspect Software while maintaining much of the debugging session context
 - Create, modify, and monitor register as watch items

The following features were added to the Visual Inspect Version 2 Software.

Update 2.4

- Support for debugging DLLs. With Visual Inspect 2.4, users can
 - Read snapshot files created from processes with DLLs
 - Suspend program execution when a DLL is loaded using a new event breakpoint (DLL Open) so that they can set breakpoints in initialization code and debug the code as well
 - Continue or abandon a step that was in progress when a process was suspended because of a DLL Open event breakpoint
 - Exclude or include DLLs in the set of linkage units that you want to debug symbolically
 - See the name of the DLL in which an external symbol was found
 - Enable/disable DLLs and other linkage units by toggling a checkbox or selecting a context menu command
- The Linkage Units Monitor now lists DLLs and can be used to enable or disable linkage units.
- A new tab (Persistence) in the Preferences dialog controls the persistence of linkage unit state (enable/disable) and linkage unit symbol file. Three new linkage unit-related warnings can be specified on the Warning tab in the Preferences dialog.

- Watch Window now has two new fields: Binding and Linkage Unit. A new Properties button on the Watch Window displays properties of the watch item selected in the Watch Window.

Update 2.3

- The new Step Continue command in the Program Control View Execution menu and Context menu allows you to complete a previously interrupted step.
- The maximum number of local variables displayed in Program Control View can be configured.
- The RUNV macro in the OSS environment now detects your workstation IP address.

Update 2.2

- Program Control View is automatically displayed.
- There are visual indicators on the Program Control View Stack Frame List indicating top of stack or return location.
- There is a Configure View capability and optional Linkage Unit field on Breakpoints Manager.
- Code breakpoints are sorted in the Breakpoints Manager.
- The Configure View capability on Watch Window allows you to display and rearrange fields.
- Watch Window items can be sorted by name, scope, and location.

Update 2.1

- Source text can be copied to the clipboard.
- Program control can be transferred to Inspect or Visual Inspect Software.
- IEEE Floating-Point is supported.
- There are C/C++ pointer improvements.

Service and support

Help

Help for Visual Inspect is integrated in the software. It can also be found online at <http://www.hp.com/go/VisualInspect>.

Education

HP offers education courses in programming languages and debugging. For schedules in your area, contact your HP representative. To enroll in courses offered in the United States, call 1 (800) 621 9198.

Support

Standard support is provided for the Visual Inspect debugger.

Technical specifications

System requirements

	Client	Server
Hardware		
Processor	Intel® Pentium® family of processors or equivalent	HP NonStop server
Memory	32 GB recommended	
Disk space consumed	8 MB	
Software		

Supported platforms, servers, and applications

Operating system support	Windows NT® 4.0 with Service Pack 6a Windows 2000 with Service Pack 3 Windows XP with Service Pack 1a	HP NonStop operating system, Mission Critical Operating Environment, Release Version (RV) D45 or later, RV G05 or later, and RV H06 or later
Communications	TCP/IP	TCP/IP subsystem
Database support	Visual Inspect Software supports both the NonStop SQL and Enscribe databases	
Languages that can be accessed	C, C++, COBOL, and TAL/pTAL All the EPIC, RISC, and CISC versions of the languages are supported	

Ordering information

Part number	Product name
HP Integrity NonStop NS-series servers	
HSB40V3	Visual Inspect Version 3.0 client for Microsoft Windows NT 4.0, Windows 2000, and Windows XP operating systems Visual Inspect servers ship standard with the NonStop operating system, RV H0x or later Visual Inspect client (HSB40V3) is available at no charge when purchasing HP Enterprise Toolkit—NonStop Edition (HSB93V3). Just add HSB93P1 (promotion ID) and HSB40V3 to your Enterprise Toolkit order.
HP NonStop S-series servers	
SB40V2	Visual Inspect Version 2.4 client for Microsoft Windows NT 4.0, Windows 2000, and Windows XP operating systems Visual Inspect servers ship standard with the NonStop operating system, RV D4x or later and RV G0x or later Visual Inspect client (SB40V2) is available at no charge when you purchase HP Enterprise Toolkit—NonStop Edition (SB93V1). Just add SB93P1 (promotion ID) and SB40V2 to your Enterprise Toolkit order.

For more information

For more information about HP Visual Inspect Software, visit www.hp.com/go/nonstop.

HP Financial Services provides innovative financing and financial asset management programs to help you cost-effectively acquire and manage your HP solutions. For more information on these services, contact your HP sales representative or visit www.hp.com/go/hpfinancialservices.

HP Technology Services provides a broad spectrum of services to commercial and enterprise customers. In addition to HP hardware and software support packages, HP Technology Services also offers performance and availability services, proactive mission-critical services, and services ranging from deployment to support management of the entire IT infrastructure, including HP and multivendor environments. For more information on these services, contact your HP sales representative or visit www.hp.com/hps/support.

© Copyright 2005 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

For more information, visit www.hp.com/go/nonstop.
05/2005

