



MAP QUEST

BSM REQUIRES THOROUGH KNOWLEDGE OF FAILURE POINTS WITHIN AN APPLICATION INFRASTRUCTURE. WE TESTED FIVE DEPENDENCY MAPPING TOOLS ON OUR NWC INC. NETWORK; RELICORE'S CLARITY WAS, WELL, CLEARLY SUPERIOR



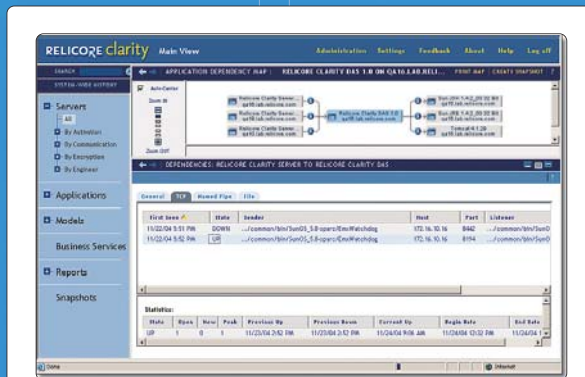
To be successful at BSM, you must map your apps. How else will you know which IT failure is mucking up your business service? We asked for tools that collect systems configuration data, decode packets and watch kernel I/O to determine application dependencies. These are not service, performance or status monitors—rather, they map all your servers, systems, applications, processes, services, users and sometimes even network devices to decipher what applications depend on, and what depends on them. This mapping also makes diagnostic, capacity and service management application delivery easier.

For our tests, we used our NWC Inc. business applications network. Of course, we know exactly how NWC Inc. works, but we expected the

BY BRUCE BOARDMAN

products to document it all as if we didn't. If you're wondering whether it wouldn't be simpler, and cheaper, to use the documentation generated by your application architects, developers or administrators, remember this is the real world. In all but the smallest infrastructures, those records are sketchy. When IT must deliver many applications using hundreds of servers, manual documentation, synchronization and auditing are nearly impossible.

We invited eight IT service mapping and monitoring vendors to participate in this review. We told them our primary focus would be discovering application-



Relicore Clarity provided the most granular dependency maps out of the box of all the products we tested.

service dependencies; products that perform only network infrastructure monitoring and correlation weren't eligible. Answering the call were BMC Software with BMC Topology Discovery (TD) 1.2, Cendura Corp. with Cohesion 3.5, Collation with Confignia 3.1, nLayers with InSight 4.0 and Relicore with Clarity 4.1.

Mercury Interactive and Tideway Systems declined. Integrien Corp. sent its Integrien Alive, but the product's exceptional performance-management functionality doesn't include application mapping.

Draw Me a Picture

Application-mapping tools are an automated source of reliable topology data for determining the impact of service outages and changes—vital functionality for BSM initiatives that seek to link IT infrastructure performance to service-level management targets based on business objectives. Reliable maps also can shorten MTTR (mean time to repair) by revealing how an app is running in production. And this data benefits IT capacity and change planners by identifying specific modules, executables or files that may need to be upgraded, while highlighting which dependencies exist to determine how resources are likely to be affected. Production control can audit planned changes and automate updates to a CMDB (configuration-management database).

These are not cure-alls, however. Although some of the data collected by application-mapping products can further server-performance-monitoring objectives, for example, you'll do better with a dedicated performance-management suite, like those from OpTier or ProactiveNet. And, as noted, we didn't require entries to map network dependencies; some switch and load-balancer information will bubble up when using the tools tested, but if you need Layer 2 and Layer 3 maps

of switch and router connectivity, look to products like Hewlett-Packard's HP OpenView Network Node Manager and Computer Associates' Aprisma.

We tested application mapping by using the business processes established in our NWC Inc. business applications lab in Green Bay, Wis. The application-mapping servers—and the tester—lived in our Syracuse University Real-World Labs®; we connected the two environments over a SonicWall VPN. (For more information, see "How We Tested BSM Tools," page 40.)

Our widget-manufacturing operation consists of three main applications: an e-commerce site, an inventory site and a customer-order-tracking site. We mapped all three applications. Inventory and e-commerce share a single Oracle9i database, while the Microsoft SQL customer-tracking database is synchronized with the Oracle database. Each application has Web server, application server and database tiers, though the app and Web servers live on the same machine.

How It Works

One misconception about application-mapping software is that it will provide a high-level view of application-service delivery—the realm of vendors like Computer Associates, Managed Objects and Smarts. Rather, these discover detailed configurations and device-application relationships. For example, they told us what servers are running what applications and with which databases and clients they communicate.

Of the products we tested, Cohesion, TD and Confignia take a configuration-based approach, while InSight and Clarity look at transactions and I/O. Of course, audits performed on configurations are valuable. However, looking at transactions and I/O is more inclusive, and this approach has two distinct advantages over scan-

REAL-WORLD LABS

REPORT CARD

BSM Tools

	Relicore Clarity 4.1	nLayers InSight 4.0	Cendura Cohesion 3.5	BMC Software BMC Topology Discovery 1.2	Collation Confignia 3.1
AUDITING CHANGE (25%)	4	4	5	2	3
DEPENDENCY MAPPING (25%)	5	4	3	4	3
DISCOVERY PROCESS (20%)	4	4	4	4	4
ADMINISTRATION (10%)	4	4.5	4	4	4
PRICE (10%)	3	4	3	5	3
REPORTING (10%)	5	5	4	2	4
TOTAL SCORE (100%)	4.25	4.15	3.90	3.40	3.40
<small>A>4.3, B>3.5, C>2.5, D>1.5, F<1.5 A-C GRADES INCLUDE + OR - IN THEIR RANGES. TOTAL SCORES AND WEIGHTED SCORES ARE BASED ON A SCALE OF</small>					
<div style="display: flex; justify-content: space-around; font-size: 2em; font-weight: bold;"> B⁺ B⁺ B C⁺ C⁺ </div>					
<p>AUDITING CHANGE grades depend on how well products detected changes in application environments, such as file and registry entries.</p> <p>DEPENDENCY MAPPING rates how well the products documented the dependencies a specific business service has on an application and an application has on processes, files, OSs and servers.</p> <p>PRICE scenario is broken down by 100, 1,000 and 10,000 monitored dual-CPU servers, plus annual license/maintenance as percentage of purchase.</p> <p>Customize the results of this report card using the Interactive Report Card®, a Java applet, at www.nwc.com.</p>					

ning configurations. First, it helps those who want to upgrade or retire systems by revealing which apps and files are being used. We also like documenting unknown applications or services, like an unknown DNS server. Second, products that passively watch I/O put less stress on servers than those that actively scan for file, registry and directory changes. Also, passive monitoring leads to more complete results than scanning does, with the caveat that if a device isn't used, an I/O approach won't see it. See "Mapping: Count the Ways," page 47, for a rundown of approaches.

Our grades are based on dependency mapping, auditing change, the discovery process, reporting, administration and price. We were especially interested in dependency mapping, and here Relicore Clarity's process granularity helped it nudge out BMC TD and nLayers InSight. Clarity showed processes communicating through applications to other servers, rather than relying on well-known TCP/IP ports.

These three products also delivered useful depictions of relationships. TD and InSight offer the best graphical representations, but we weren't just looking for pretty pictures. These products are behind-the-scenes players, providing data to CMDBs and service-mapping products. Thus, how well-detailed dependencies are mapped is more important than drawing them graphically, so again Clarity led the way, clearly showing our systems, ports, applications and processes communicating in real time.

Cohesion and Confignia scanned our system configurations and applied built-in knowledge of, for example, SAP or IBM WebSphere applications. We could expand their knowledge bases to incorporate custom applications. The resulting system snapshots yielded detailed configuration-state information, which Cohesion and Confignia pieced together to create topologies. All the products use configuration information, but only Cohesion and Confignia rely on it for creating application dependencies. For example, every product discovered

HOW WE TESTED BSM TOOLS

We performed our tests in our Syracuse University Real-World Labs® using applications in our NWC Inc. business applications lab in Green Bay, Wis.

Discovery typically started with finding the NWC Inc. subnet, but often we would create groups of target servers representing the three NWC Inc. apps we used for testing: e-commerce, inventory and order tracking. The e-commerce application, built on Web services and running on Oracle application servers with Oracle Apache Web servers, are load balanced by an F5 BigIron. A back-end Oracle9i database, the application and Web servers run on Windows.

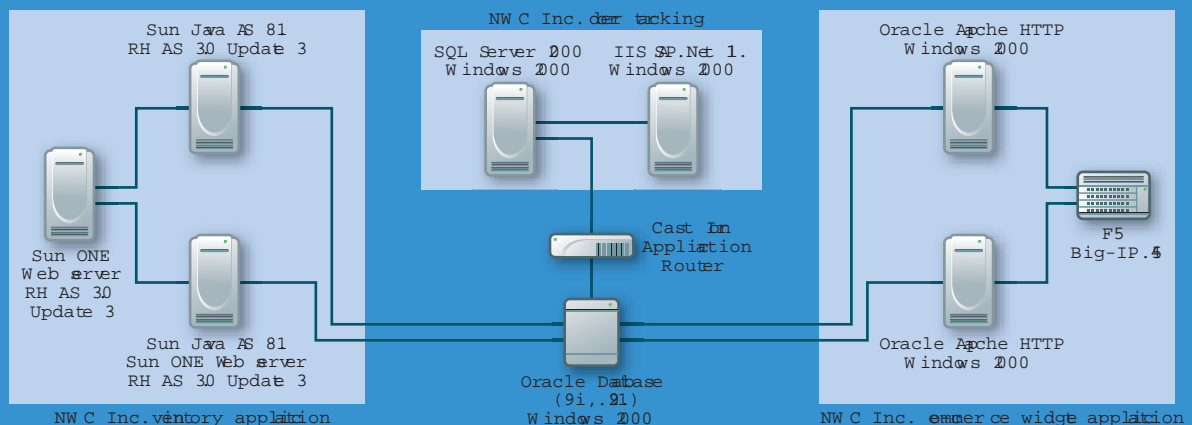
The inventory application runs on Sun ONE Web servers and Sun Java application servers, all running Red Hat Enterprise AS 3.0. The inventory application talks to the same Oracle9i database being updated by the e-commerce app. The order-tracking application runs on IIS ASP on .Net talking to a Microsoft SQL Server 2000 database. Both servers run on Windows 2000. The Microsoft SQL Server is synchronized with the Oracle9i database using a Cast Iron application-integration appliance.

We set up probes, agentless access and agents on all the servers. We ran transactions automatically and interactively into all of the applications during discov-

ery to enhance the connections that Clarity, InSight and TD could infer. For all the products, we defined administrative and/or root privileges to all servers.

A shout of thanks to Lori MacVittie for providing and babysitting the NWC Inc. application test environment.

All NETWORK COMPUTING product reviews are conducted by current or former IT professionals in our Real-World Labs® or partner labs, according to our own test criteria. Vendor involvement is limited to assistance in configuration and troubleshooting. NETWORK COMPUTING schedules reviews based solely on our editorial judgment of reader



Apache running on our e-commerce server. But the products that scanned configurations relied only on the files and registry entries they found on the hard drive for this data, not what was loaded and executing.

InSight, like Clarity, creates application maps based on real-time data collection, but instead of hooking into our kernel I/O, it sat on a tap or span port, watching traffic like a probe. InSight also has an agentless data-collection option. In addition, InSight maps apps and application processes through a combination of gathering traffic and agentless scanning of TCP/IP ports and is generally protocol-centric, while Clarity watches I/O at the OS kernel using a proprietary agent and reports on what processes are communicating across, as well as within, systems.

In tests, Cohesion also placed an agent on our monitored server, but it culled through system configuration data matching executables, files, parameters and registry entries to templates that identify and track applications. BMC TD and Confignia used agentless collections to gather configuration data, then matched it to application templates.

Other Testing Factors

All the products audited for changes in app environments and provided file and registry entries, but Cohesion went further by taking snapshots of detailed configuration data, down to our process parameter values,

giving us a clear ad hoc display of changes. One nit with all the products tested is that creation of groups is a rather tedious manual process—we had to select by application or configuration component, not by server.

Configuration differences were generally emphasized between system snapshots and different systems using a graphic key and highlighting to show changes, adds and deletes. We couldn't compare different OS configurations because the data isn't normalized—you'll have to interpret and analyze syntax manually. Think of it as job security.

To score pricing, we asked vendors the cost of monitoring 100, 1,000 and 10,000 dual-CPU servers plus annual license/maintenance as a percentage of purchase. For the 1,000 server scenario, pricing ranged from \$75,000 to a high of \$750,000 for Relicore's Clarity, but Clarity fell in the middle of the pack for 10,000 servers. Support hovers around 20 percent (see a full breakdown in our features chart, page 48). Bottom line, much will depend on the size of your installation, and most vendors indicated a willingness to negotiate. BMC's TD seems inexpensive, but that's because it's part of a larger suite and is lacking some functionality. As always, worth is relative: The value of automatically mapping application configurations is easily proved if an application outage is shortened or avoided. For more on where BSM makes sense, see "The Network Effect," page 30.

BSM TOOL VENDORS AT A GLANCE

PUBLIC COMPANIES

Company name	Year founded	Product name	Year launched	Market capitalization as of Aug. 10 \$ 000	Other products	News
BMC SOFTWARE (BMC)	1980	BMC Topology Discovery 1.2, BMC Topology Discovery Expert Extensions for SAP, Siebel and J2EE applications	2004	\$4,360,000	DealerTrack, Medco, SwissCom, Toyota Motor Group	Garnered license bookings of \$33.6 million for service management in fiscal Q1 2006, second only to the company's mainframe management bookings of \$34.7 million

Source: Company reports, Yahoo.com

PRIVATE COMPANIES

Company name	Year founded	Product name	Year launched	Employees	Key customers	News
CENDURA CORP.	2002	Cohesion 3.5	2005	75	Amica, BlueCross, Cars.com, Digital Insight, Germania, GM, Homestore.com, HSBC, Intuit, Medtronic, Quintiles, Sybase, Verisign	Announced integration between Cohesion and Hewlett-Packard Network Node Manager on Aug. 8
COLLATION	2001	Collation Confignia 3.1	2002	29	Federal Express, Nielsen Media Research, SanDisk, The World Bank	Announced support for IBM IT Service Management family in May
NLAYERS	2003	nLayers InSight 4.0	2004	32	BCBS, Interwoven, Museum of Modern Art, Netvision	Closed \$6 million round of Series B financing in first half 2005
RELICORE	2000	Relicore Clarity 4.1	2002	50	Allmerica Financial, Cisco Systems, Fidelity Investments, Hughes Network Systems, JP Morgan Chase, Starwood Hotels & Resorts Worldwide, UniGroup	Hired new CEO in first half, replacing co-founder Firdaus Bhathena with former Willy Technology exec Victor Nyman

Source: Company reports

Clarity earns our Editor's Choice award for having the best dependency mapping, even though its agent requirement may drive some to our second-place finisher, Insight. We believe Insight will provide the fastest ROI of the products we tested, thanks to its relatively reasonable price and easy implementation, and it earns our Best Value award.

Relicore Clarity 4.1 Relicore Clarity provided unmatched application-process granularity and deter-

B+ mined which of our NWC Inc. components, applications, services and servers were communicating. But its big advantage is also its main disadvantage: To do its hoodoo, we had to place an agent on every server, where each monitored the kernel I/O—this is a great place for dependency gathering, but a tough and touchy place to camp out. We didn't experience server stability problems, but agents can be organizational and political hot potatoes.

Installing the agents was a simple matter on both Windows and Linux systems, though on Windows we had to reboot—this is a Windows rather than a Relicore issue, but it still sucked. Once agents were installed, the application quickly drew granular dependencies automatically using canned models to identify applications and component connections. This is similar to the other products' method of relying on "blueprints" or other preconfigured application configuration and communication knowledge to recognize applications out of the box. All the canned models can be cloned to create new, slightly different models.

We tested using some custom NWC Inc. applications. First, we used a wizard to build recognition for our custom app. We chose to define particular application components from a representative server, then selected specific executables to identify the custom application. We did need to know what processes or services uniquely identified the custom application, but given that, the process was point and click.

The UI designates server, application, component and business service group views of dependencies. But instead of four separate views, these are UI entry points organized by group. Once a device or application was displayed, we moved from the component to the server and saw application and server dependencies.

Clarity presented our business services as a grouping of discovered apps, which we used to connect the servers in each of our NWC Inc. apps logically. We could include applications, components and other business services in an overall or "different service" view, which is what we did for NWC Inc. We created separate business views for each application—e-commerce, inventory and order tracking—then nested these into a single business view.

The organization of Clarity's dependency maps was unique among the products we tested. Instead of multiple layout options (for example, tree or hierarchal), the display was exclusively left-to-right hierarchal. The object, server, application, component or business service we selected anchored the hierarchy. Then, to the

FYI

BSM Merger On April 7, Computer Associates announced it would acquire Concord Communications. Gartner advises that those considering Concord's system agents, event correlation, application response measurement, and business service management products wait until CA offers a

right, the dependencies were displayed in greater detail, representing outgoing communications. Devices to the left of the selected object are incoming dependencies. For example, we chose the business service "NWC Inc.," which included all of the NWC Inc. business services, constituent applications, servers and components, and NWC Inc. was the left-most object. The next level right displayed the e-commerce, inventory and order-tracking applications. Moving one more column to the right, the server, application and components that comprise those applications were displayed. Mapping continues this way to the process, which is the lowest level of dependency displayed.

Missing from the Clarity map—and something the other products took pains to display—were relationship



lines that describe parent-child and connection directions. TD and InSight did a good job graphically laying out these relationships, but Clarity placed the system or applica-

tion in a center focus of a block diagram, then showed incoming dependencies (those in which the connections from other systems and applications were relying on the focused element) and outgoing dependencies (those in which the focused element is a client).

Most of the systems we tested provided comparisons of executables and OS versions, which was helpful in determining what had changed. Clarity takes this a couple of steps further, comparing components of services and their dependencies. We could compare as many as 12 servers (a display limitation) against snapshot versions and against a gold copy to identify configuration drift.

Snapshot creation was granular. We created snapshots of entire servers, similar to snapshots Cohesion and Confignia offer, but Clarity also let us select a single application module for a snapshot. In addition, Clarity let us collect baselines and apply them to thresholds. Baselines can be for any period of time up to a year but will primarily show basic minimum/maximum/average percentiles or dynamic thresholds. The system search shows changes over time, so if a problem happens it's easy enough to pick a time frame and get a list of everything Clarity knows has changed.

Although the other products tested recognize apps by a combination of well-known ports and application configuration details, Clarity's kernel point of view means it sees processes communicating without knowing anything about them. Clarity still needs application knowledge to identify the application, but even lacking that, application-process granularity will still be discov-

ered. We'd like to know more about how it does what it does, but this is clearly in the realm of secret sauce.

We were a bit nervous about allowing agents into our kernels, but Relicore told us its agents generally run at less than 1 percent of CPU usage, though that could rise to 3 percent on systems experiencing a high level of change. Memory requirements are about 50 MB. Relicore says that the memory and processor requirements on agentless systems are comparable. We didn't see much utilization difference among the products.

The Clarity UI is intuitive without a lot of excess, and we could save our favorite views. Moving from server to application to process to port was a breeze, and no matter where we started in the interface, we could find what we were looking for. We especially liked the flexible view feature, which let us create views with the column index and sort values that made sense. Both TD and InSight beat Clarity in the administration category, thanks to the drag-and-drop and right-click context launch capabilities available in their interfaces.

Clarity's pricing was simple—each server is an increment, tiered for higher volume. Our scenarios ranged from \$1,250 per server at 100 servers to \$100 per server at 10,000 servers plus 20 percent annual license and maintenance, and Relicore stressed its willingness to negotiate.

Clarity 4.1. Relicore, (781) 229-1122.
www.relicore.com

nLayers Insight 4.0 We performed discovery with the InSight appliance using both active mode and its passive discovery mode, which was unique among the products tested. For passive discovery we sat the appliance on a span port or tap and began decoding network traffic. Data is collected in real time, displayed in near real time and stored for historical usage baselines. In addition, InSight uses an active agentless approach to log onto Unix systems using SSH, to Windows over WMI (Windows Management Instrumentation) and SNMP for both.

We ran across a couple of errors during discovery. A few server addresses and names were combined in a way that didn't make sense. The version we were running can link two servers with the same IP address, but in our case these were different machines that for an unknown reason both erroneously listed in their SNMP MIBs an *ipAdEntAddr 0.0.0.0* value. The company says the new version, which we didn't test, adjusts for this anomaly.

In addition, NAT (Network Address Translation) and encrypted traffic were a problem for InSight. NAT shows up as connecting clients and servers to whatever the NAT device is, as long as there are probes on both the public and private sides of the NAT. If data is collected from only one side of the NAT, the connections terminate on the NAT device. Encrypted traffic is, of course, opaque to passive collection and a primary reason for InSight to combine active discovery.

We could view dependencies in InSight graphically or in table form, at the level of a service on a port, and

within an application running on a server communicating with another client or server. The graphical view includes historical data on the number of connections between devices. So, for example, clients and servers communicating within our NWC Inc. application were shown as lines, with varying widths depicting number of connections and arrows indicating dependency.

We created groups defining the IP addresses of the servers we knew belonged to our NWC Inc. transactions. InSight doesn't require grouping to find what's happen-



ing on the network, but this method removes unnecessary clutter from the display and clarifies what an application is doing. A neat option let us dynamically add clients and

servers we saw talking to our defined servers. Our group definition initially was whatever default services InSight chose, but changing these is no biggie.

Different types of group views are available. We set up a business applications view to represent our three NWC Inc. applications. This gave us a container that included the core servers for each. A cool feature in the views let us make a separate group that we could then populate from a business-application view. In this way, not only did we avoid redefining our business group, we extended this new view to include clients and servers. Nice.

We explored InSight's dependency mappings using its filtering functionality. When we asked to view our defined NWC Inc. application, only InSight was nice enough to warn us that there were umpteenth connections that would make for a messy map. Throwing caution to the wind, we plunged into the spaghetti mess and used the filtering function to turn off select protocols, like DNS, LDAP and unclassified. Once we removed everything except HTTP and Oracle transactions, we could clearly see the client and back-end dependencies.

InSight also linked to the underlying protocol showing our client-server HTML, Web page objects and rudimentary load. It's not a performance-monitoring application, but we appreciated seeing the data collection that led to specific dependency mappings.

One method for auditing changes uses InSight's connection baselines. We set a policy that said any time a connection to the Oracle server was more than 10 percent of the baseline, we'd get an e-mail. This is a crude performance-status mechanism and requires an understanding of baseline fluctuations, but baselines can be relearned at any time for any device. To be useful, performance data must be collected and compared week over week, day over day and hour over hour, at a minimum, and in our experience this sort of static baseline control works only with relatively static usage. However, this capability does make use of the data collection already in place and ensures that huge changes in usage won't go unnoticed.

InSight is easy to set up and provides a quick pay-back, displaying application mapping quicker than any of the other products, though its probe-based passive

collection through packet inspection required footprints on all our segments. Application-dependency information was surprisingly detailed—InSight discovered a DNS server and all the clients relying on it.

Setting up the InSight appliance and data-collection probes was trivial as well. The interface is intuitive, with most functions only a couple of clicks apart. Upgrading the server and managing probes is done from within the UI. It took us a while to get the hang of navigating and executing the various features and functions, but once we got in the groove, most functions were three clicks away. We used only a single probe for testing, but, of course, for every segment monitored, another probe is required.

InSight's built-in reporting is divided into three categories: inventory, dependency and components. Each has canned reports on hosts, services and applications. We could run reports using business applications and views as sort criteria, and reports are displayed as PDFs and exportable in Excel and RTF formats.

InSight's second-lowest price was a surprise. We figured on a much higher cost given the level of dependency mapping and additional reporting and audit features, especially since our pricing scenario forced nLayers to add in probes for every 24 servers, but it came in at a mere \$40 per server for 10,000 devices. And, InSight is an even better deal because low implementation hurdles will provide a quick ROI.

InSight 4.0. nLayers, (877) NLAYERS, (408) 554-7470. www.nlayers.com

Cendura Cohesion 3.5 Cendura Cohesion provided good change auditing and documented dependencies

B for our applications and application components on a periodic basis. We gathered application-configuration knowledge using an agent. Cohesion then compared our gathered data with its extensive baked-in knowledge about how applications

FYI

Sell It on eBay? Through 2006, market hype will drive 30 percent of enterprises to try business service view tools. But just 10 percent will have the change-management discipline to successfully maintain and update the tools. The remaining implementations will become shelfware, according to Gartner.

are configured and what they rely on. Our biggest gripe was the difficulty in understanding and applying the process for applications that aren't supported out of the box. Luckily, the supported applications list is extensive, and includes BMC WebLogic, IBM WebSphere, Oracle, PeopleSoft, SAP, Siebel and Tibco.

Dependencies are shown as a map, similar to the what the other products provide, with lines drawn between servers and applications. Both a tree view and graphical asset view of application components are available. We found the tree view easier for displaying complex components, such as Java Web applications, SOAP applications and databases, because we could expand each to see subcomponents.

In our test bed, Cohesion presented two layers drawn with multiple active lines, which when selected popped up a property chart that explained relationships in terms of what host or application was configured to connect. Dependency maps are as of the last scan or snapshot.

Cohesion's dependency maps contained complete lists of processes, executables and files for our custom applications. But more isn't always better—we spent time removing application-documentation files and extraneous logs using Cohesion's method for defining applications by building a blueprint—essentially the configuration footprint that identifies a particular application. Blueprints were as accurate as we were willing to make them. For our simple apps this process wasn't difficult, but as complexity or similarity of custom applications increased, more blueprints were necessary.

Cohesion's main weakness compared with rivals was a flat graphical dependency map for our custom apps. Missing were displays of systems containing applications that contain processes. Instead, systems were simply shown to be communicating. On the bright side, dependencies between components on different systems were documented. When we chose a connection between two hosts, for example, a text box with "Reference To" and "Reference From" displays associated files, executables and parameters between the connected systems.

Cohesion makes extensive use of application snapshots to compare variations of hosts, applications and application components. Snapshots are created on demand or on a defined schedule. We set our applications to be scanned twice daily, but any frequency is technically doable. We ran comparisons against snapshots of the configuration in our database as well as the current configuration as it existed on our network. The output—table or tree—displayed changes and explained which configurations had been altered based on which application image we chose as the source. Analyzing

THE NWC INC. BUSINESS APPLICATIONS LAB

in Green Bay, Wis., provides the NETWORK COMPUTING technology team with a real-world environ-

ment for testing complex, enterprise-class software. Our networks and servers function as production systems and maintain databases that house financial, inventory, customer and order data to support our fictional widget manufacturer. Our evolving ADS and Exchange implementations maintain a list of employees and access rights. NWC Inc. also operates a thriving online business supported by an Apache, Oracle and Microsoft Web infrastructure.

See the lab live at <http://192.155.11.50>, read about our day-to-day operations at www.nwc.com/blog/archives/nwc_inc/index.html, and order some widgets at inc.gb.nwc.com (our favorite, Widget

returned data was easy, with differences delineated as being available only in the source (the image chosen first) or only in the target (the image chosen second); if a parameter changes, both are displayed.

Cohesion also let us check for specific values. In the blueprinting process we specified required and default values, then entered a bogus value for a J2EE parameter and ran the compliance engine against a set of NWC Inc. applications we'd created. Up went a flag.

Out of the box, Cohesion offered quite a few suggestions for default compliance checks that could serve as best practices. It advised us to remove example scripts from an IIS server installation, for example, and eliminate anonymous access from an FTP server—maybe obvious, but still useful. Configuration checks are assigned a low, medium or high weight, so minor misconfiguration noise doesn't mask important mistakes. Given tuning, these checks could be a reliable diagnostic and admin tool.

Cohesion uses agents for application discovery but scans the network or imports DNS to get an initial group of devices. Once our devices were contacted successfully, we installed agents remotely from the console on those machines hosting applications we cared about. It took a couple of scans to get all the target machines, but installation was painless, and we didn't have stability problems. However, we frequently found after making a snapshot and checking our blueprint that one or two agents were listed as not having responded. A demand poll usually brought forth a successful connection. We suspect time-out and retry values needed tweaking because of the WAN connection between server and agents, but we couldn't tune those parameters.

But that's not the end of discovery—we needed blueprints to identify our applications. Cohesion uses the concept of an application consisting of multiple components, such as specific Web, application and database

servers. At first this seemed arcane, but we went ahead and ran the process, adding components to our blueprints and observing what was installed on the servers in question. The upshot? This process is not perfect. It requires tuning and learning the guts of blueprinting, and at the close of its scan of target servers Cohesion coughed up a results page asking us to input things like database names, sysadmin user IDs and passwords, and the instance of the database to be included. It would have made sense for the databases found to be instantiated—after all, we had already supplied access credentials.

Cohesion's reporting comprises a dozen canned reports that range across inventory, audit, compliance for apps and hosts defined to the system. The reports are delivered as HTML, CSV or XML but run only interactively, not as scheduled tasks. Report options can be saved as custom reports, but ad hoc report writing is lacking.

Cohesion's pricing was next to lowest for 100 servers but the highest when monitoring our 10,000 server scenario. When estimating the cost at the high end, Cendura said for that kind of deal, negotiated prices are the norm.

Cohesion 3.5. Cendura Corp., (866) 263-5500., www.cendura.com

BMC Topology Discovery 1.2 BMC's Topology Discovery is part of the larger BMC suite designed for



BSM. As such, TD produces an excellent mapping of application dependencies but lacks service management and change auditing.

This reduced the cost of TD but also hurt its score. On the other hand, compared with rivals, TD did a good job showing and documenting what applications rely on and the impact of those relationships.

Different levels of communication and dependency exist within TD. For example, the discovery process

MAPPING: COUNT THE WAYS

There are three basic approaches to mapping application dependencies:

» **Passive.** A network node watches sessions and decodes packets to determine which servers are talking; which protocols they are using; and, either by knowing application transactions or through port identification, which applications the servers are supporting. Probes are used to gather traffic passively through a network tap or mirror port, and actual communicating applications are used to build dependencies.
Pros: There's no system footprint

and unknown servers are found. Also, easiest to implement.
Cons: You can't see application-process communications, and implementation at aggregation points is required because traffic monitoring is limited to the local segment.

» **Agentless.** More precisely, proxy-agent approaches log on to servers using SSH, telnet, WMI, JMX and SNMP to gather configurations. They compare gathered configurations to predefined executables and files that identify the application.
Pros: There's no footprint on the

systems and you can see application processes and files. This setup represents a good trade-off between agent and passive approaches.
Cons: You can't see application-process intercommunications and servers must be accessed.

» **Agent.** A small, proprietary piece of code is placed on each server being monitored. The code looks at configuration, I/O and performance of the server, sucking some cycles and memory in the process.
Pros: The agent sees all processes and sometimes sees inter-

automatically finds dependencies using configuration and open ports to determine connections, but we also set static dependencies that remained between discoveries. In addition, connections were defined between applications, within application processes, and between servers and clients. For example, our e-commerce application and Web servers ran on the same box. TD

showed the Apache processes for both, then connected to each separately. This eased visually connecting each process to its application within the physical server. And, applications communicating over third-party processes, such as databases communicating through a listener, are separated from communications between servers. We also could see connection directions.

BSM Tool Features

	BMC Software BMC Topology Discovery 1.2	Cendura Cohesion 3.5	Collation Confinia 3.1	nLayers InSight 4.0	Relicore Clarity 4.1
Data collection method agent/agentless/passive	N/Y/N	Y/Y/N	N/Y/N	N/Y/Y	Y/Y/N
Optional discovery modules	Y	N	N	Y	N
Planned upgrades	Mainframe z/OS, storage, Microsoft environment (Active Directory, Exchange, .Net), MQ Series, Tuxedo, Tibco, BPEL standard, Colaxa, BEA, WAS and Oracle apps	Solaris 10, RH Linux 4.0 AS, WebSphere 6, Oracle 10g AS, Suse Linux 9 and 10, Windows 2003 64-bit	Web service engines, WebMethods, Oracle Financials	SAP Netweaver, Oracle Applications	Oracle Human Capital Management, Application Server 10g, Customer Relationship Management; Peoplesoft Enterprise Financial Management, Enterprise Human Capital Management; Microsoft Active Directory Service, Exchange Server 2003, SQL Server 2005, IIS 6.0 Oracle 10g Database
CMDB integration	BMC Remedy AR System, BMC Atrium CMDB	Cohesion database is CMDB	Netcool RAD, Tivoli TBSM, Compuware Vantage	InSight Federated CMDB, with SDK CLI export	Relicore Clarity CMDB, HP OpenView Service Desk, IBM Tivoli Discovery Library, Peregrine Asset Center
Product data storage	Embedded Oracle 9i database, external Oracle, MS SQL, Sybase	Sybase, Oracle, MS SQL Server	Confinia provides a full-fledged CMDB; existing CMDBs can also be populated with data collected by Confinia	Ships as a self-contained appliance and is built on an industry-standard Oracle RDBMS	Oracle 9i, IBM UDB 8.2
Projected vendor sizing*	5,000	1,000 - 2,000	2,500	30,000	1,500
Third-party integration	SAP, Oracle, IBM	HP OpenView, IBM NetView	Micromuse Netcool RAD, Compuware Vantage, IBM Tivoli TBSM, BMC Remedy, Comconsult AixBOMS, Tivoli TEC, Nagios	IBM Tivoli Discovery Library TBSM and TEM, Tivoli OPAL program, Tivoli Provisioning Manager; Managed Objects BITI, Bladelogic Operations Manager for Data Center Automation	HP OpenView, HP OpenView Service Desk, HP OVO/W, HP NNM; IBM TBSM, Tivoli Provisioning Manager, Tivoli Discovery Library, Tivoli Enterprise Console, Tivoli NetView; Managed Objects Formula, Peregrine Asset Center
Price					
Monitoring 100 dual CPU servers	\$53,000	100,000	\$150,000 + maintenance	\$150,000	\$125,000
Monitoring 1,000 dual CPU servers	\$75,000	Starts at \$300,000	\$300,000 + maintenance	\$150,000	\$750,000
Monitoring 10,000 dual CPU servers	\$360,000	Starts at \$2 million	\$1.25 million + maintenance	\$400,000	\$1 million
Percent of purchase for annual license/maintenance	20%	18%	18% for 8/5, 22% 24/7	18% standard, 20% extended	20%
Training available/included	Y/N	Y/N	Y/N	Y/N	Y/N

Y=Yes, N=No * Sizing scenario: Server with four 3.2-GHz CPUs, 4 GB of RAM and onboard SCSI RAID 5 storage doing twice-daily scans that collect application-level metrics. Required database

FYI

ITIL Rising 68 percent of 400 IT professionals surveyed by Enterprise Management Associates are currently using a process model to improve their IT service management; of that

We created a dependency map that focused on our central Oracle database server. Then, through TD's Dynamic Navigation feature, we selected a database process that was listed as unknown. The navigation feature found four server processes and added them to the map. The granularity isn't as deep or as frequently updated as that in Relicore's Clarity, but TD does provide process-level granularity on which processes are talking.

We found dependencies based partly on actual communications and partly by the software knowing configurations and ports. All the diff reports were well-designed, and though they lacked syntax or analysis, it beat having to compare our apps manually. TD refers to qualifying a dependency before it becomes viewable, meaning it is assured that the dependencies depicted are accurate. During our test they were. Dependency maps can be exported to BMC CMDB or a third-party CMDB, but only qualified dependencies will be included. Export is supported over HTTP, but BMC says a SOAP interface will be supplied in its next release.

TD is apply named—it has the most complete control over discovery and map population. We were supplied with a nicely engineered wizard that stepped us through the discovery process. We could specify what we wanted to discover in assets, applications and network categories. Assets refer to system configuration data found by remote (agentless) queries using SSH and WMI. TCP Port scans are also used to determine the apps and services present on the network.

We liked the process of displaying systems discovered by Universal Application Discovery because, rather than having to define specifics of executables and TCP/IP ports, the browser-like "Dynamic Navigation" feature let us explore tasks and processes. It's easier and more intuitive than Confignia's custom definition, which had us go outside the discovery process to explore system executables and TCP/IP ports to define the application, then run another discovery.

BMC performs audit and configuration management in a separate product, Remedy Change and Configuration Management, not included in this review. However, TD's dynamic navigation discovers and attaches processes to executables as they exist. So while not auditing against a snapshot or externally defined set of application-configuration parameters, TD still gave us a thorough data collection of specific apps and associated running processes.

TD is easy to install and use, once we got the hang of the interface. For every function—and there are a lot of them—there are a couple of entry points. BMC pointed out that TD includes a one-week training course. The price range—\$53,000 to \$360,000—is the lowest of all the products tested, but TD provides only

application dependency mapping—albeit good application dependency mapping.

BMC Topology Discovery 1.2, BMC Topology Discovery Expert Extension for SAP, Siebel and J2EE applications. BMC Software, (800) 841-2031, (713) 918-8800. www.bmc.com

Collation Confignia 3.1 Confignia scans systems, applications and files against defined application configurations; as with Cohesion, some configs are included, and custom applications can be defined. When matched, the relationships suggested by the configuration parameters define how application dependencies should be. The downside of this approach is missing applications that are running but not defined, but once the application definition matches to all the running applications, a good production control process will help ensure accuracy.

Confignia's UI is well-designed and walked us through the discovery process, which included setting targets by IP and then scheduling, defining access and setting up custom applications as needed. Confignia supports generic and Cisco network devices in addition to WebSphere, WebLogic, Domino, Sybase and Microsoft SQL database servers as well as LDAP and generic Windows and Linux servers. Unfortunately for our testing purposes, Oracle and Sun ONE were not supported, so we had to define custom servers.

Confignia displays hierarchies of devices and applications using lines. When we selected a device or single service, upstream and downstream services were highlighted. But we were unable to connect processes automatically to apps within servers without completely defining those relationships. Once defined, Confignia discovered the apps and processes and surpassed TD, Insight and Clarity by helping us attach unknown found processes. Confignia's diff report compared applications, services, subnets and OSs, listing parameters in a table format and using red to highlight when things didn't match.

Confignia includes audit and status reporting. Our components and applications were tracked for change or drift from a known-good version, and a report highlighted dormant components, servers and applications that had not shown up in a discover scan for a while. Helpful for picking up after yourself.

Confignia's price range of \$150,000 to \$1.25 million came with the proviso that your mileage will vary, especially at the high end. Its unique pricing increments are on a per-server basis, with two CPUs equaling a server.

Confignia 3.1. Collation, (650) 587-3300.
www.collation.com NWC



BRUCE BOARDMAN, executive editor of NETWORK COMPUTING, tests and writes about network and systems management. He has 12 years' experience managing networks and distributed computing for a financial service provider. Write to him at bboardman@nwc.com.

Post a comment or question on this story at