Maximizing Portal Application Performance

Modern Portals Require a Modern Portal Performance Management Solution
Overview
Enterprise Portal Servers are vital cogs of today’s Web Applications, providing a single interface point for personalized access to information, applications, services, and business processes. Advances in portal technology have significantly reduced the time required to develop and deploy portal applications. More organizations are leveraging enterprise portals to improve developer productivity, customer service, architectural flexibility, and application integration, essential requirements for achieving rapidly evolving business objectives. A well-performing portal adds immense value to the IT organization and the line of business. Conversely, poor performing portal applications can lead to budget overruns, schedule delays, lost business and lower customer satisfaction.

As portals move into the mainstream side of enterprise integration applications, maintaining high performance and availability is becoming a key initiative not only for operational measurements, but also for the overall return on investment of the technologies, themselves. Unfortunately, the technology advancements that helped create development productivity gains have added layers of complexity, some of which are “invisible,” making it more difficult to monitor and manage portal application performance. In this white paper, we will explore key portal technological advancements, their impact on the ability to manage performance and how ClearApp’s QuickVision Portal Manager can help enterprise organizations overcome these management challenges.

Key Advances in Portal Technology Accelerated Enterprise Adoption
Application owners have always adored portals, even before mainstream portal servers were available. Large organizations could build their own portal enablement frameworks. Of course, this was a costly proposition, requiring organizations to fund large development, QA and support organizations for a non-revenue project. The introduction of enterprise portal frameworks built on J2EE platforms helped facilitate industry-wide rollout of portal applications. The introduction of standard frameworks for development and deployment of UI elements has led to easier, faster, cheaper application projects.

Thus, more organizations are deploying portal-based applications. Additionally, each individual portal project tends to integrate a greater number of individual applications in any given project – with most organizations planning and executing multiple portal application projects at once. The current level of technology has the capabilities to enable almost any IT shop to easily and quickly build and deploy complex applications. New integrated functionality enabling massive acceptance includes single sign-on security services, built-in business process design and management, plug-in search capabilities and integrated content management systems. With the ease of design, deployment and administration, organizations are investing the time, energy and dollars to convert their Web Applications to Portal Applications across the entire enterprise.
**Efficient reuse and improved flexibility**

With enterprises making strides into Web Services and Service Oriented Architecture (SOA) applications, enterprise portals add immense value. For example, the Web Services for Remote Portlets (WSRP) standard provides efficient reuse and flexibility by distributing the deployment of portal resources across the infrastructure, enabling easy development of federated portals that pull remote content and business functionality into a central web UI. The loosely coupled nature of WSRP also enables enterprises to quickly add and modify portal functionalities to best achieve their business objectives.

**Modern Enterprise Portals Present Unique Management Challenges**

While technology advances have ushered in the age of federated portals, they have also created unique management challenges that conventional application management tools and techniques can’t overcome. Like many IT Operations challenges, some customers are actively trying to fix the pains associated with these issues, some feel they just have to deal with the pain, and others are still unaware of the problems (until that day the application crashes and costs thousand or millions of dollars). The unique challenges of managing enterprise portal applications are discussed in the following sections.

**Challenge 1: Usage context of portal components impacts performance management**

Modern portal frameworks and products make it easy for enterprises to create new portals by quickly assembling discrete portal components (i.e. pages, tabs, labels). This set of entities defines how pages function and how each user’s individual page is configured. This collective set of items represents the Portal’s Functional Logic and represents the first management challenge in Portal application performance for several reasons:

- The Virtual Portal defines the available entities for any particular URL, but cannot capture how each user has personalized their specific pages
- Since Functional Logic components are not J2EE code, J2EE monitoring tools can’t measure nor understand the performance of these entities
- Thus, a management gap exists since neither the deep dive J2EE monitoring solutions nor the outside-in URL-based solutions can measure the performance of the Virtual Portal – or even understand how transactions flow from one (URL) to the other (J2EE)

Finally, the immense re-use of portal application components (such as portlets), requires that the business transaction context should always be considered when taking measurements. For example, if a Stock Quote portlet is used by 3 different functions across multiple pages, the performance should be individually and collectively monitored.

Up to now, some organizations have tried to perform the correlation from URL to deep application measurements with a manual process, creating and manipulating dashboards that artificially capture the presumed relationships. This worked enough in simple portal applications to warrant its continued use.

With more component sharing, distributed applications and Web Services, today’s complex federated portals have pushed past the ability to manually correlate and artificially create the business context through all calls and all components in both the Virtual Portal and the application code, itself.
**Monitoring Business Logic and Functional Logic**

The application code (J2EE or other platform) sitting behind the customer-facing portal User Interface is commonly referred to as the application business logic. This layer of the application manages the integration of multiple data, messaging and transaction systems to deliver the appropriate data or transactions to the end user. However, end users *drive* the portal functionality through the portal’s integrated UI framework, made up of entities such as Books, Tabs, Labels and Pages. Whether utilizing a customized version of the UI or the standard layout, end users execute the business logic through a layer of abstraction tied to the portal UI framework functionality. This layer of logic that creates the UI and ultimately drives requests into the business logic is referred to as the **Functional Logic**.

![Image of a portal with tabs and labels](image)

An Application’s Functional Logic is visible on the Web Page in the form of Tabs, Labels, Pages and Portlets (such as *My Weather* and *My News* above)

Monitoring and managing a Portal Application’s Functional Logic is as critical, if not more so, than monitoring the Business Logic. The importance of the Functional Logic and its performance stems from the fact that the Functional Logic is the closest representation in the application layer of the actual user experience, and the only way to tie performance to *how* users are functionally using the application.

One challenge of monitoring the Functional Logic is understanding just what is there, what the entities are, how they’re laid out and how they work together to create the functional logic. The next challenge is being able to tie the high level Port functionality to the business processes that use it – *and* to the underlying application components that make up the business logic behind the portal.

But an ever greater challenge exists in highly distributed, highly shared application components (like Portals usually are). Shared components are management nightmares due to the many different ways a particular component (or portal entity, such as a Tab) is called and used. Monitoring the true performance of the entire portal application requires *each and every* measurement to be taken within the context of the calling business process, transaction, portal entity or application component. A portlet that is used by 3 different pages should be measured 3 different ways (in addition to its overall aggregate metrics).
Challenge 2: Fast changing nature of portals requires management updating
Since portal applications are usually deployed as an effort to address more frequent changes to business requirements around the application functionality, dealing with constant environmental change is critical when considering application performance management. As enterprise portals are quickly assembled, modified, disassembled and reassembled, the task of tracking these changes and updating the associated management environments becomes more difficult, creating an interesting dilemma: APM owners can either manage generically, turn monitoring off completely or resign themselves to investing weeks of manpower and outside consulting for each and every application change – not exactly a great set of solutions.

Challenge 3: Federated portals with WSRP adds management complexity
One interesting new portal technology advancement, the use of Federated remote Portals, enables application architects and developers to create federated portals using remotely distributed portal components. The remote components are gathered together at runtime and presented to the end user in a single web UI called the consumer portal. Unlike a local (or non-federated) portal, these remote components can be independently maintained, updated, and released without changing the consumer portal which uses them, resulting in less risky portal projects.

Switching production applications from non-federated portals to federated portals is a paradigm shift for most organizations, but the benefits are significant. Without federated portal architectures, any change – even a small change – to a single portlet would result in a complete cycle of application deployment including quality assurance, performance testing, certification, staging, deployment, management configuration and regression testing of the entire application and application environment. By distributing the business logic through remote portlet calls and a federated consumer, IT organizations only have to test the updated functionality, since everything else (other portlets and the consumer portal) remained the same. The follow diagram depicts this paradigm shift from non-federated portals to federated ones.

![Federated Portal Diagram](image)

Management tools have to keep up with this benefit, but so far, only ClearApp’s capabilities match up to the new testing process of federated portal application updating. Traditional APM tools require that any monitoring of the application that may touch the
updated portlet – even though it is remote – has to be revisited to insure that the manually created Dashboards, Custom Metrics, Alerts and Correlation are still valid. Even if they are all still okay, it could take weeks of time, manpower and consultant fees. ClearApp’s modeling engine will detect the change in the remote portlet, analyze the change, and incorporate any new information into the overall model for the Federated Portal Application. In addition, the direct correlation from the remote portlet with the federated portal application(s) that call it is automatically updated by QuickVision – including Service Level Dashboard roll-ups, drill-down links, architecture maps, etc.

While it takes less effort to deploy and maintain federated portals, the use of WSRP presents a unique management challenge for portal owners. Since the execution of a remote portlet occurs on the WSRP producer server, performance measurements of the consumer portal only provide a portion of the data needed to solve any performance bottlenecks or availability issues. To help triage performance problems associated with federated portals, management tools must include measurements across the consumer / producer gap – including the exact relationships between the consumers and producers. As an application user more web services, create more distributed applications and employ massively re-used components, the complexity creates difficulty in the support team being able to humanly understand what all the relationships are between consumers and producers.

\[ 
\text{Producer 1} \quad \text{Producer 2} \quad \text{Producer 3} 
\]

\[ 
\text{Consumer Portal} 
\]

**Intelligent Automation Can Overcome Portal Management Challenges**

Manual tactics are not optimal for dealing with these complex management challenges. A time-consuming manual approach that requires extensive manpower and trial-and-error (thus, being error prone) cannot scale as needed when trying to manage more complex applications, more distributed applications, or simply more applications. Intelligent automation is the only way to overcome the unique challenges associated with managing portal applications. In this section, we will explore ClearApp’s automated approach with the QuickVision management solution.
Automated configuration significantly reduces time, effort, and errors
There are three key steps in the Application Performance Management lifecycle:

1. **Setup and Configuration**
   - Discover all applications, Business Logic components (such as EJBs), Functional Logic components (i.e. Pages) and all relationships between the components
   - Build Maps of all possible paths through the Functional and Business Logic
   - Making a discerning selection of the appropriate application components to monitor for complete coverage of the business processes

2. **Correlation and Analysis**
   - Track relationship between each application component and all the business processes that use them
   - Take all measurements within the context of the top level business function and/or process: *Connect the Portal Functional Logic to the Application Business Logic*
   - Automatically rolling up alerts and warnings to each successive level (method → component → service → application → server → portal → process)
   - Provide both **drill-down** capabilities for root cause analysis and **drill-up** capabilities for business impact analysis of specific problems

3. **Change Detection and Management**
   - Detect changes to any aspect of the application environment, including new / updated java classes, brand new or updated applications, new servers, etc.
   - Automatically analyze how the changes impact the overall transactions (i.e. select new metrics, deprecate unneeded measurements , map new transaction paths)

---

**THE CLEARAPP SOLUTION**

**Step 1:**
QuickVision automatically finds all applications in the designated IT infrastructure and then looks inside each one and starts modeling

**Step 2:**
QuickVision's modeling engine automatically maps out all component relationships—inside and out—adding transaction paths, call paths, and business logic into the overall model

**Step 3:**
QuickVision uses its knowledge of call paths and transaction paths to specify what (and how deep) to monitor: then creates a customized management console that automatically correlates component-level metrics to the transactions, applications and processes

**Step 4:**
QuickVision detects changes in real-time, updating monitoring and management configuration as needed without any human intervention. Thus QuickVision is always managing the right data at the right level at the right time.
How a performance management solution deals with changes is critical to the continued success of APM projects. In a distributed / SOA / Web Services application, even a small number of services can mean dozens of changes every month – sometimes even in a week. Manually adjusting the setup, re-correlating metrics and re-configuring the service level monitoring is untenable, and leads to degradation of the APM solution’s value.

**Consistency Leads to Control**

As organizations roll out more applications, more complex applications and more distributed applications (especially portals and federated portals), the manual approach to application performance management configuration and maintenance becomes untenable. Organizations find themselves making an unenviable trade-off – spend massive amounts of money and time to constantly analyze, reverse engineer, manually correlate and update each and every application in the environment or accepting significantly less value from their APM solution.

An automated approach is the only way for an APM solution to effectively handle the complexity of distributed, framework-oriented applications, especially over time as changes continue to hit the application, infrastructure, configuration, etc. By automating the three steps outlined above, Operations and Portal Administration teams can:

- Quickly deploy service level management across broad application portfolios, with correlation from the business process level down to the critical methods deep inside the application business logic
- Deliver consistent coverage and reporting capabilities, regardless of the personnel involved in the deployment and/or configuration of the APM solution
- Handle change in the application environment automatically, without requiring intervention, or even analysis, from application development or architecture experts, maintaining the consistent excellence of the APM solution

For federated portals that use WSRP, QuickVision not only selects the appropriate tracer locations, but also tracks and displays WSRP consumer-producer relationships. This unique ability to dynamically monitor WSRP consumer-producer relationships enables QuickVision to provide other unique capabilities such as automatic organization of performance measurements and topological visualization.
Contextual measurements, visualization, and organization improve management
QuickVision is also the only APM solution that accounts for the usage context for each and every portal entity (Books, Tabs, Labels, etc.) and application component (Servlets, EJBs, JSPs, etc.) for each and every request automatically. This capability enables QuickVision to determine how a shared portlet performs for each of several different portal pages. With an automatically generated map of all entity–component relationships, QuickVision can measure this portlet performance in each calling context (response time and call counts for each and every unique page / label / tab / … combination that calls it. The following diagram illustrates this concept.

In the example on the right, Portlet E is used by all three portals. To truly manage the performance of Portlet E, any measurements must be made with an eye to context of where the request originated (Portal 1, 2 or 3). QuickVision’s unique ability to track usage context yields contextual measurements that appropriately breakdown Portlet E’s behavior by different portals.

Where conventional APM solutions will report the overall performance of Portlet E, a complete selection of metrics includes the following:

**Response Time**
- Average, Min, Max Response Time of all requests to Portlet E
- Average, Min, Max Response Time of all requests to Portlet E from Portal 1
- Average, Min, Max Response Time of all requests to Portlet E from Portal 3
- Average, Min, Max Response Time of all requests to Portlet E from Portal 3

**Request Count**
- Number of Requests and Completions of Portlet E for all requests
- Number of Requests and Completions of Portlet E for requests from Portal 1
- Number of Requests and Completions of Portlet E for requests from Portal 2
- Number of Requests and Completions of Portlet E for requests from Portal 3

Any other metrics taken about Portlet E (concurrent requests, error counts, etc.) must also be measured with an eye to overall transaction context.

Additional benefits of continuous contextual monitoring is the ability to report on high level service levels for all transactions and processes through the measurement and analysis of low level application components. And the service level measurements will be
more accurate since no reported metric will be skewed by the performance of requests from other transactions, pages, labels, tabs, etc.

Of course, ease of use is still tantamount to making an APM solution as effective as possible, so QuickVision includes an unparalleled User Interface designed to allow anybody in IT Operations to report on service levels, monitor performance and perform the necessary levels of triage and root cause analysis when problems occur – without having to learn anything about the application structure, architecture, or environment.

QuickVision’s portal layout view, called the Virtual Portal Desktop™, enables IT application support teams to look at an enterprise portal from the perspective of the end users. The following screenshot illustrates this capability:

Each Portal Framework has a set of descriptors for its page rendering logic (the Functional Logic), called meta-data. QuickVision knows how to extract this metadata and translates the portal application’s functional layout into the ClearApp Virtual Portal Desktop™, a patent-pending visualization of the portal functional entities such as pages, books, labels, tabs and – of course – portlets. The QuickVision can bring up the performance of a page/book/label combination by traversing the Virtual Portal Desktop (VPD) the same way that an end user would traverse the Portal Page, itself.

The combination selected in the VPD results in request count and performance information about the particular combination. In the picture above, for instance, the Self
Service Portlet on the Customer Service Page in the MyAccount Book has been selected (note the Orange outline on the Self Service portlet). The performance pane on the right has the performance of all the different calls to the portlet. Note that there are actually two different ways that the SelfService portlet is executed, both of which show up in the performance preview pane.

All the relationships between the Portal Page entities (discovered from the metadata), the portlets that get executed and the application business logic, as well as the correlation between the application code components and the overlying portal entities that use them, are all determined automatically by QuickVision – with no human intervention, programming or configuration required.

Shared Components
Understanding performance and availability of shared applications with regard to the specific functional calls they’re serving is critical in distributed applications, or applications that contain many shared (or re-used) components. A shared component could be an Enterprise Java Bean, Servlet, JSP or a portlet. To examine this, let’s consider the same application we’ve been working with. There is a Login to Portal portlet that performs the same types of authorization and authentication functions for multiple portal processes. The performance of the portlet may be different based on the specific portal functions that are being requested.

In the above screen snippet, notice that the Login to Portal portlet actually has three distinct measurements, one for each of the three pages that uses it – providing performance measurements within the context of the actual user request – at all times.

As noted in the previous discussion on contextual monitoring, QuickVision always shows associated usage context data to create the most logical organization of performance metrics. In the above screenshot, a person diagnosing the application’s worst performing portlet (Case Management) can use the associated contextual information about portal, desktop, book, and page to quickly locate the problematic Portal and Transaction node.

Analytics and diagnostic views accelerate performance problem resolution
While intelligent automation, contextual measurements, and logical visualizations enable application support teams to effectively monitor portal with minimal time, effort, and expertise, QuickVision provides additional automated capabilities around performance analytics to accelerate performance problem diagnosis and resolution.

To accelerate problem identification, QuickVision contains powerful features like URL Query. URL Query allows any IT Operations or Support personnel to paste a URL address from the end user into QuickVision. QuickVision automatically determines the...
Portal entities executed by the URL and the underlying application components used by the request.

Once a potential problem has been identified, the application support team can bring up the appropriate diagnostic view to begin root cause analysis. An easy way to perform quick triage is to use QuickVision’s unique application architecture view – a diagnostic view that enables users to look at active call path and low-level performance measurements.

In the above QuickVision screenshot, the left pane (Application Architecture View) shows the relevant active call path of the end user’s request. The right pane (Delay Analysis) shows the most egregious users of time and resources in the transaction call path. For the specific transaction analyzed above, the Delay Analysis identified that 70% of the overall call path delay was contributed by a single component, and EJB called LD.Entity.PropertyManager.

Performance analytics like Delay Analysis help guide QuickVision users to performance bottlenecks quickly. QuickVision users can click on any identified problem component and retrieve additional method-level and SQL statement-level measurements for even deeper problem analysis. With these performance diagnostics and analytics features, IT Operations and Support personnel reduce the time and effort required to resolve problems.
**Conclusions**

Today’s enterprises are deploying portals to address ever-changing competitive business needs in a flexible highly fluid application environment. While advances in J2EE portal technology have given more enterprises the ability to deploy complex portal applications, these organizations are now experiencing challenges in delivering and managing the performance of these applications. These new challenges require production portal management solutions that understand both business logic and functional logic, accurately track and measure usage context, automatically keep up with application changes, and effectively deal with the added complexity of Federated portals with WSRP.

ClearApp’s QuickVision Portal Manager is the only APM solution for J2EE Portals that can overcome the unique challenges of managing the performance of today’s portal applications. QuickVision leverages its patent-pending modeling technology (called AppSchema™) to automate the three key steps of Application Performance Management:

- Setup and Configuration
- Analysis and Correlation
- Change Detection and Management

ClearApp’s automated approach significantly reduces the time, effort, and expertise required with the setup and maintenance of portal management environments, saving upwards of hundreds of thousands of dollars in total solution costs. The modeling engine also drives the unique ability to detect, understand and manage the performance of the application Functional Logic. And ClearApp’s unique visualizations and maps build off the AppSchema model to provide an easy-to-use User Interface for any IT personnel to monitor performance and triage problems when they occur – without requiring extensive knowledge of the application, itself.

Modern enterprise portals need modern portal management. Join enterprises that have already deployed QuickVision and overcome portal management challenges with intelligent automation.

To learn more about ClearApp’s products and technologies, please visit:

http://www.clearapp.com
SUITABILITY NOTICE

CLEARAPP, INC. AND/OR ITS RESPECTIVE SUPPLIERS (TOGETHER, "CLEARAPP") MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THE INFORMATION CONTAINED IN THE DOCUMENTS AND RELATED GRAPHICS FOR ANY PURPOSE. THE REPORT AND RELATED GRAPHICS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND.

CLEARAPP HEREBY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS INFORMATION, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL CLEARAPP BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DIRECT DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTUOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OF OR PERFORMANCE OF INFORMATION IN THIS REPORT. REASONABLE CARE HAS BEEN TAKEN IN THE DEVELOPMENT OF THIS REPORT. ACTUAL CUSTOMER USE RESULTS WILL VARY FROM REPORT RESULTS. THE REPORT MAY INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CLEARAPP MAY MAKE IMPROVEMENTS AND/OR CHANGES TO THE REPORT OR THE SOFTWARE AT ANY TIME.