Emerging J2EE Challenges:
Maturing J2EE Application Support

A META Group White Paper
Created for Veritas
## Contents

Executive Summary ................................................................. 3  
Introduction .................................................................................. 4  
  The J2EE Choice........................................................................... 4  
J2EE Operational Challenges ......................................................... 5  
  Immature or Failing Applications .................................................. 5  
  Lack of Internal Knowledge .......................................................... 5  
  Lack of Ownership....................................................................... 5  
  Poor Problem Resolution Process .................................................. 5  
Emerging Organizational Changes .................................................. 6  
  J2EE Engineering......................................................................... 6  
  Availability Managers.................................................................... 7  
  Application Owners....................................................................... 8  
Actions .......................................................................................... 9  
Bottom Line .................................................................................. 10
Emerging J2EE Challenges:
Maturing J2EE Application Support

Executive Summary
Doing business on the Web is important to future strategy and success, and underlying much of the successful technology on the Web is Java — specifically the Java 2 Enterprise Edition (J2EE) application environment (in use by at least 80% of the Global 2000). However, most companies are struggling to properly support this environment. Current efforts are full of lack of ownership of the production-level support of J2EE technology, poor communication to business executives, and sporadic, uncoordinated investment in management tools.

To get a grip on these environments, META Group recommends several steps. For addressing the organizational issues, companies should seek to fill the following roles:

- **J2EE engineering**: This should be a shared or semi-shared service, much like a database administrator (DBA). This role is responsible for the technical support of all underlying J2EE technology (e.g., application server, virtual machines).
- **Availability manager**: The goal of the availability manager is to link business’s perception of availability with the IT organization view of availability. An availability manager becomes the “go to” person for the business when a business interruption is perceived — as opposed to business managers dealing with numerous discrete IT roles.
- **Application owner**: An application owner is responsible for all aspects of an application (or limited set of applications) in a production environment. While they do not necessarily provide the day-to-day technical support, they are responsible for the overall coordination of all activities around the application (e.g., change management, configuration management, interfacing with service-level management, interfacing with requirements gathering).

The challenge is that all of these roles require data to make decisions in relation to J2EE applications. Therefore, companies should seek a toolset that can cross the boundaries of these groups. If the individuals involved in support do not share data, the environment is set to fail. Even if all the roles are in place, if there is no measurement for them to work with and no warning sent, no results can be expected and user satisfaction will suffer. It is important to seek broad monitoring that can cross technology boundaries (e.g., J2EE environment through database) and provide drill-down analysis for deep problem resolution.
Introduction
It is clear that doing business on the Web is important to future strategy and success. No matter what the industry, every company is shifting at least some function to a Web-based application. A driving factor is the ability of Web-based service delivery to reduce overall cost and enhance the user’s experience. META Group has seen organizations that realize savings in the millions, no matter what the currency. By reducing complexity through the leveraging of Web-based business service delivery — through automated check-in at airports, online banking, and internal Web-based human resource or expense tracking systems — companies see not only savings in technology costs but also a reduction in business process costs. The technology behind these savings has increased in significance and criticality and is now vital to business execution.

For these reasons, META Group finds no major company in the world that is not executing on a Web strategy. Behind the scenes of Web delivery are Web-based applications. Yet there is a fundamental change underway in how applications are created. Traditionally, the business would outline the extensive requirements for all the desired features and functions a new application would need to deliver. Then the information technology group would develop said application and spend the majority of effort on implementing these details, with little concern for what would happen once that application was up and running in production.

Now, however, the focus is moving away from extensive features and functions toward the operational characteristics (e.g., speed) and support of a production application. Basically, ensuring that an application performs well in production and is available as often as it should be are now the key elements of the development process. Since these applications face the ultimate customer or have significant revenue and business implications, companies cannot risk failed technology. Organizations are now delaying development of some features and functions to ensure that a supportable application can be deployed, which is a winning strategy. Once the application is proven, it is then updated and enhanced rapidly to enable all the desired functionality. However, along the way, the IT organization is rated on the combination of supportability and functionality.

The J2EE Choice
Although there are choices in what technology is used for development of Web-based applications, META Group research indicates that 80%-85% of the Global 2000 organizations have at least some Java-based application in use. Specifically this is through use of the Java 2 Enterprise Edition, which Sun Microsystems describes as defining the standard for developing component-based multi-tier enterprise applications. J2EE applications are typically implemented via an
application server. META Group views the J2EE application server market as consisting of Java-based application servers or “containers” that manage components and services in a multi-tier application. The application server and associated libraries of services have become the primary platform for most new enterprise application development, providing the core infrastructure once supplied by an operating system.

**J2EE Operational Challenges**
With widespread use of J2EE as the foundation for the important Web-based business services and the need to support them in production, companies must understand the following key issues in supporting J2EE environments.

**Immature or Failing Applications**
With the majority of large J2EE applications in production for less than two years, it is clear that the applications are not fully mature. As a result, companies have reported that availability of Web-based applications is not acceptable. The most common problems range from code issues (e.g., bugs) to failed communications between the many complex technology layers (e.g., application server communicating with the database). These issues are complex to identify but must be worked out in order for companies to take these applications to the next level.

**Lack of Internal Knowledge**
Companies have not yet invested in building enough internal knowledge to support these applications. Most of the knowledge currently belongs to developers. This puts developers into the uncomfortable and unfamiliar position of production support. This is not an optimal situation. It is widely understood that the J2EE environment is complex (Source: http://www.eweek.com/article2/0,4149,1122575,00.asp), which means that investment in skill development is required. For every other significant technology within IT (e.g., databases, storage, servers), a skill set is invested in and developed. J2EE requires the same.

**Lack of Ownership**
Along with lack of knowledge is ambiguous ownership of the J2EE technology in production. Walk into any company and ask who owns a J2EE-based application or the J2EE application server driving that application, and the response is likely to be silence. There is no clear plan and no consistency across companies, which of course leads to chaos.

**Poor Problem Resolution Process**
Clearly, with a lack of knowledge and ownership, processes for dealing with failures, slowdowns, or even rollouts are poor to non-existent. Even when a
company does invest in a tool for monitoring of its application (which META Group recommends), the tool is deployed in a haphazard manner and often the value expected from it is not attained.

**Emerging Organizational Changes**

When analyzing the key challenges, it becomes clear that most companies are not properly organized to handle J2EE applications. When people are in the proper places, they often do not have the tools they need. To deal with the problem META Group has identified three new roles emerging within the IT organization that impact the support of a J2EE application: availability managers, application owners, and J2EE engineers.

Of course, there are also the traditional IT roles playing a part. Business relationship management or even the business owner of a given application must press for the technology to be supported properly. This need will be best met by demanding service-level agreements. Application developers and quality assurance analysts are most concerned with delivering the correct functionality, and rightly so.

However, with the rapid rate of change normally present in a Web-based application (something enabled by the object nature of J2EE), successful organizations find developers more aligned with the production side of IT. This however must be an alignment, not ownership. Neither the development team nor the QA team is well equipped to deal with major failures or to understand the complexities of a production enterprise. Also, the operations centers (e.g., data center operations, network operations, application operations) all play an oversight role in the support of these applications. The activities of these groups must be coordinated and they must work off the same data (often fostered by sharing a common management tool).

**J2EE Engineering**

Companies must address the low-level need to manage the J2EE application server. An owner of the Web application server and related technology directly responsible for maintenance and support of these elements is required. J2EE application servers are just as complicated as databases, servers, or other middleware (e.g., MQSeries). Therefore, J2EE application servers require the same engineering support that those technologies receive. The role of a J2EE engineer or J2EE administrator must be created as a shared service, or semi shared service, similar to the database administrator role. DBAs are assigned to a subset of databases, usually with related applications. They do all the low-level
detail work in maintenance, problem resolution, and planning associated with those databases, which are the same tasks for the J2EE engineering role.

Also like a DBA, the J2EE engineer would interface with development. This interface would be to diagnose problems and also participate in upcoming development planning. This role would work with application owners and, where necessary, availability mangers, but with a much lower level of responsibility.

The J2EE engineers will be the ones drilling deep into problems. Instead of escalating Level 2 or 3 support to development, this engineering team will assess those issues and determine what needs to go to the development group, if their team is unable to assist. The end result will be a logical entry point into production for developer relationships, reduced production support effort from the application teams, and eventually a better operating J2EE application, resulting from dedicated attention.

Creation of this role would be the critical first step in building internal knowledge in the production side of the IT organization and alleviating some pressure from development or other groups that have been picking up the slack.

J2EE engineering requires detailed diagnostic tools. Although individuals in this role do care about the end-to-end measures, they are most focused on understanding the internal aspects of the J2EE engines. They are not focused on availability, but on overall performance (e.g., “How does our current heap level affect overall performance?”). Investment must be made in low-level monitoring tools that enable the J2EE engineering team to gather the best information, or the position will be less effective then intended.

Of course, some diagnostic tools require excessive effort from the user. Many companies are now implementing management tools to seek specific anomalies or unearth problems they know they are having but cannot pin down. When seeking tools for J2EE engineering companies must assess the tool’s ability to easily identify not only detailed data but also how much embedded intelligence exists, which will enable automated identification of problem areas with little prior knowledge of its existence.

**Availability Managers**

The goal of the availability manager is to link the business’s perception of availability with the IT organization’s view of availability. The role of the availability manager — that is, becoming the “go to” person for the business when a business interruption is perceived rather than business managers dealing with numerous discrete IT roles — is new, which means that titles and reporting structures are
inconsistent. Yet it is clear that many organizations are investing in creation of this role. Nearly 10% of the Global 2000 currently have the availability manager role formally in place (growing to 20% by YE04), with an estimated 20%-25% operating the function in an informal manner (growing to 45%-50% by YE04).

Since J2EE applications are most commonly those that face external users and customers, they become the first applications of interest to availability managers and are also the applications that the services business managers most care about. When an availability manager is put in place, business users know whom to call on when a perceived problem occurs, which simplifies problem resolution. Availability managers will not know the details of how to resolve the problem or even how to identify its presence, but they will be aware of the business impact and criticality. They will have an impact on the decision of how many resources must be put on a particular problem and how much a company will be willing to invest in resolving or preventing problems with this application in the future.

Availability managers require high-level tools. They do not need to have details, such as those provided by “dig in” products that give low-level data. Instead, they are most interested in the end-to-end point of view — how the user or business service is impacted. Availability managers must be given utilities that show the users’ perspective of health and provide enough information that they know which engineering or support groups to call on when a problem emerges. Business view consoles, end-to-end monitors, and tools that integrate data from multiple measurement points are most likely to be useful.

**Application Owners**

Application owners are responsible for all aspects of an application (or a limited set of applications) in a production environment. They are not responsible for the underlying infrastructure (e.g., servers, network, storage, even databases) but are responsible for all the application-specific software, for coordinating changes, for working with availability managers on any issues, for interfacing with architects to ensure standards are met, etc. They become the “go to” person for all the key aspects of a single application or a set of applications.

All J2EE applications require an application owner. These applications are very complex. Even an application server, which is communicated as a simple component, is a complicated, multilayer, multicomponent set of software elements. Application owners may need to understand these components, but more importantly, they must understand how the components work together to serve a specific application. They must know what the impact of changing one element will be on another.
The application owner role is responsible for identifying that problems are actually bugs and for pushing the development team to get fixes in. Application owners are responsible for enhancing availability if it is not meeting business needs and are also responsible for coordinating application changes to minimize impact. J2EE applications change rapidly, and without this role in place, there is no one to proactively ensure that changes do not introduce instability.

Application owners will also leverage tools. Their needs are very similar to those of an availability manager, except that they care about more than just availability issues. They also care about performance issues (e.g., trends toward a slow down), capacity planning issues, and identification of new targets for monitoring. It is logical for application owners and availability managers to share a toolset, but they will each use it slightly differently.

**Actions**

IT organizations must act now, or they will risk their critical business functions not having proper support. META Group recommends that companies:

- **Create (at a minimum) a J2EE engineering role:** This role should be implemented as a shared service, allowing the knowledge to be leveraged across multiple applications. This is critical to building deep internal knowledge on supporting J2EE applications.
- **Create application owner and availability manager roles (if resources allow):** This role should provide oversight to these critical applications and also provide better interfacing to business groups that are relying on these technologies.
- **Monitor the J2EE environment:** One cannot improve what is not measured, nor can failures be detected. As noted earlier, even if all the roles are in place, if there is no measurement for them to work with and no warning sent, no results can be expected and user satisfaction will suffer. Broad monitoring must be put in place that can cross technology boundaries (e.g., J2EE environment through database) and provide drill-down analysis for deep problem resolution.
- **Invest in a shared toolset:** One common element among all roles in IT is that they require data. There is great difficulty if all roles work off different information. If each group uses its own tool to monitor the same technology, different data is viewed and different conclusions may be reached, which may lead to divergent action. Sharing tools means shared data, which increases communications as well as the number of people working on the same problem, which leads to faster resolution. It is crucial to ensure the deployment of these tools and their true usage, or the investment is wasted.
Bottom Line

J2EE is here to stay. Companies will increase their reliance on this critical technology and must begin to plan to operate it with the proper level of attention. This will involve creation of new positions in the ITO to handle the complex needs of all J2EE consumers, from low-level technical audiences to business executives.

A key element of success is the sharing of data across the groups involved in J2EE support, typically through use of shared management tools. Tools should be sought that can help multiple audiences (e.g., developers, the testing team, operations group, business executives), with correlation and intelligence embedded in the tools. The ultimate goal is true-end to-end monitoring, with clear lines of ownership and all groups working off the same data and making coordinated decisions.

Corey Ferengul is a vice president with Operations Strategies, a META Group advisory service. For additional information on this topic or other META Group offerings, contact info@metagroup.com.
About META Group

*Return On Intelligence*™

META Group is a leading provider of information technology research, advisory services, and strategic consulting. Delivering objective and actionable guidance, META Group’s experienced analysts and consultants are trusted advisors to IT and business executives around the world. Our unique collaborative models and dedicated customer service help clients be more efficient, effective, and timely in their use of IT to achieve their business goals. Visit metagroup.com for more details on our high-value approach.