Thanks for attending this presentation on The Cutting Edge of Web Applications...How WebSphere's Next Generation Application Server Brings New Value to *Your* Business. Regardless of where you find yourself along the continuum of e-business adoption, WebSphere provides you with the ability to grow and adapt for competitive advantage. This session will be of interest to you if you are looking for a high performance, scalable, and standards-based infrastructure; if you need to realize high return on investment, and you have a constrained development budget; or if you need to quickly respond to constantly changing business and market conditions.

What you will see by the end of this session is that WebSphere provides the foundation that offers superior investment protection through unparalelled integration of existing enterprise applications within an e-business environment; enhanced return on investment by providing an exceptionally productive application development environment; and a highly flexible and adaptable approach to unlock the advantages of Web services and dynamic e-business.

So enough of the introduction, let's lay out a few key themes as a framework for our discussion.

At developerWorks live, IBM began to tell the story for WebSphere Application Server V5. There were four key themes that were conveyed...and you'll see these reiterated throughout the presentation.

The first theme is about improving time-to-value by building new integration-ready applications that leverage existing software assets. This is about integrating existing assets easily into new J2EE applications (and that integration idea is represented by this handshake). We also need the ability to modify business applications easily, to compose and choreograph services and modify application behavior dynamically through things like rule engines. And that's conveyed by this sign post that suggests you can easily change directions. In a word, this is about flexibility. Integration, and flexibility.

 Another theme tied to V5 is integrated application development. The idea here is to maximize return on investment and decrease labor costs with superior developer productivity and the openness of a "portal-like" workbench that supports best-of-breed tools.

And then we see something called intelligent end-to-end optimizations. This is about superior performance (represented by the screaming locomotive), availability (represented by the "day and night" icon), and scale (depicted by the truck that can carry large loads). Optimizations are about competitive price performance, and delivering the industry's best reliability and security.

Finally, we talk about agile deployment and administration. The notion here is that you can minimize start up costs. You can start small, and you can grow to an increasingly more capable infrastructure depending on your business needs. You can retain and apply your existing skills, and you can leverage value-added frameworks and tools when you're ready to do so. Managing the infrastructure should be as simple and as seamless as possible at all points along the continuum.

Now some of the sub-themes on this slide will appear later on in the presentation as well. This includes integration, flexibility, productivity, performance, availability, and scale.

With that behind us, let's look at the session objectives.

On slide 3, we list the objectives: we're going to review the role that WebSphere Application Server and WebSphere Studio play in an e-business solution, and we're going to consider how WebSphere can bring benefits to YOUR business.

Please notice the triangle at the bottom of this slide. The point being made here is that WebSphere is fundamentally about three things. In the upper left corner you see that WebSphere is about providing personalized and streamlined access to content and collaborative services on any variety of devices. In the upper right corner you see that WebSphere is about providing business integration services both within and between enterprises. And along the bottom of the triangle you see that WebSphere provides an infrastructural underpinning for a whole range of e-business solutions. This infrastructural underpinning is described on the graphic as "Foundation and Tools"...and that's where we're going to spend most of our time in this presentation.
Let's move on to slide 4.

Slide 4 indicates that we have four parts to our agenda. We'll begin by examining the evolving role that an application server is starting to play as part of an e-business solution.
In terms of the evolving role of WebSphere Application Server, we'll bring that to life through a business-oriented case study. That's really going to be the meat of this session. Then we'll review some WebSphere foundation and tool offerings, and we'll relate some of those back to the case study. And finally, we'll consider some next steps that you might want to take on your e-business journey. So with that agenda in mind, let's get started on slide 5.

Slide 5 reviews the fundamental, classical role of application servers. This figure depicts application servers as business logic engines. Typically, application servers provide a place to execute policies; to enforce terms and conditions; and to apply business rules. Global clients commonly access these engines using browsers and pervasive devices that operate on both static and dynamic content. This content can be cached at the edge of the network to drastically reduce both network traffic and system load. Increasingly, these engines are also accessed by partners in inter- and intra-enterprise Web services interactions. Of course, the application servers don't stand alone, as they normally need to access persistent information stored in one or more databases. In addition, existing systems and applications often need to be leveraged as part of an e-business solution, and messaging technology and message brokering are frequently used to drive such legacy system interactions. We complete the picture by showing that developers need an easy way to create and update the applications that are hosted by the application server. And increasingly, business analysts need a way to model; monitor; and maintain the various business processes and policies that run there.

Moving on to slide 6, you'll see that we continue with this notion of business logic engines. The question here is, "How can vendors like IBM make these engines work better"?
You could imagine two approaches. In one case, for example the picture on the left, we might choose to *optimize* the engine with a "super oil" or "special oil" that is designed to improve efficiency and performance. The picture on the right shows that we could also *extend* an engine by enhancing its capabilities, thereby giving the engine fundamentally new utility.
Notice the fine print about optimizations and extensions. You should think of optimizations as a deployment concern. Optimizations are administration-based and they do NOT involve source code changes. An example of this would be the transformation of a standards-based J2EE application into a "turbocharged" standards-based J2EE application in which NO programming changes are required.

Extensions, on the other hand, are a development concern. Extensions DO affect the programming model, and they are frequently associated with a set of supporting development tools. An example of this would be the transformation of a standards-based J2EE application into an application that offers entirely new functionality (such as one that supports process automation services and message-to-component mapping services). This kind of extended functionality would generally be considered for incorporation into the evolving J2EE specification over time.

Slide 7 puts a little bit more meat on the bones. When we talk about optimizations, we think of things like significantly improved performance, availability, and scale. With extensions, we think about helping customers improve productivity, become more flexible and adaptable, and achieve greater levels of integration.

These optimization and extension themes have a direct bearing on pressing, cross-industry business concerns. The most common concerns that we've observed are shown at the bottom: the need to reduce costs, the need to improve customer loyalty, and the need to respond more quickly to new business opportunities -- that's what we've shown here as business agility.

Chart 6 suggests that there's a relationship between the things on the top of the chart and the issues at the bottom of the chart. For example, consider performance, which is one of the optimizations. If you could greatly improve the speed of application servers, then you could perform more work for the same hardware investment. This leads to cost savings, one of the imperatives at the bottom of the slide. Or you might consider one of the extensions, such as flexibility. If you could make applications really flexible and adaptable, for example, allowing them to implement new business policies very quickly, then you could be much more agile when it comes to seizing new opportunities and outpacing your competition. So that supports the imperative on this slide for business agility. The fact is that *each* of these optimizations and *each* of these extensions supports ALL three imperatives listed in the light blue circle.

So what should you take away from this? One point would be that application servers are growing up, they are starting to solve much bigger and broader problems than they ever have before...indeed they *need* to do this in order to demonstrate increased value within an e-business solution. It's time for application servers to take on important new responsibilities. And that's exactly what you can see WebSphere Application Server beginning to do. You'll notice on slide 8 that we're ready to put this into the context of WebSphere. So let's move right ahead from there to slide 9 where we can look at WebSphere Application Server in the context of some realistic business scenarios.

The case study that we'll consider involves a fictitious company called Web Banking Online (or WBOnline). This is a global company with international constituencies. And what are the key business challenges? The same things we're seeing everywhere: how can I cut costs, how can I improve customer loyalty, and I can rapidly adapt my business model, thriving on change, embracing change for competitive advantage?

WBOnline needs robust infrastructure, and WBOnline is fully committed to Java and to J2EE. This is vital to the case study, by the way. The *full* value of what WebSphere offers can only be realized if you have whole-heartedly embraced the Java story...

Also note the bullet indicating that WBOnline has acquired significant assets over the years. To some degree, this has come about as a result of mergers and acquisitions. The point is that

WBOnline needs to reuse what it already has, the idea of "ripping and replacing" is simply not an option.

The final point to consider is that WBOnline has some pretty sophisticated processing requirements for their application server environment. Looking at our blue bubbles from left to right, you can see that this includes the ability to interact, from Java, with multiple backend systems such as CICS and IMS. It also includes the ability to automate business processing (for example, the process of extending customers a loan). And by the way, that's about both straight-through business processing, as well as business process interactions that involve people.

What about the bubble in the lower left corner? Let's say that WBOnline wanted to apply interest credits to all of the customer accounts as part of its month-end processing cycle. This is the kind of thing that you'd want to do in batch mode. You'd want this to be a job that you could schedule for execution at some pre-determined time. You might like to have multiple accounts being updated at the same time, for example five jobs running concurrently, with each one handling about twenty percent of the accounts on record.

And then the last bubble at the bottom suggests that WBOnline wants to embrace Web services, and something called service-oriented architecture. The idea of a service-oriented architecture is that software resources can be accessed as reusable services, and this is really about ALL software resources, in all kinds of different contexts, this is not necessarily about external business partner interactions. Instead it's a paradigm shift that makes applications fundamentally easier to build, maintain, and link together. Again, we'll talk about this some more as we go through the case study together...and it's time to do just that, beginning with slide 10.

The case study kicks off with a new credit card application that WBOnline needs to build on a J2EE foundation. Here we see three requirements listed...obtain customer account information; determine if the customer is gold or above, if so issue the card; determine the credit rating, if positive issue the card.

So let's think about how WebSphere Application Server can address these requirements, beginning with some Extensions, and later on, some Optimizations.

Concentrating on the first step, we've got a situation in which we'll need to navigate through a series of CICS screens in order to obtain the customer's account data. The assumption here is that WBOnline has a so-called "green screen" application in CICS. Only by traversing back and forth through a number of screen interactions will we be able to gather up the required account information.

And then we're struck by this idea at the bottom: Why not package up these CICS interactions as a reusable service? We could hide the technical details, and all of the various complexities, behind the interface of a business service. And in fact we could use some of the key ideas behind Web services in order to define this service-oriented approach...as you'll see by moving on to slide 11...

Slide 11 has a picture of the WBOnline 3270-based screen navigations, and these are based on the J2EE Connector Architecture or JCA. But the details are hidden, and there's a simple service interface with a defined set of inputs and outputs that let the service be used by a variety of different clients in a technology-independent fashion. We're really talking about a service-oriented architecture, in which virtually any kind of software resource can be seen through the lens of a business service interface. This is all based on key Web services concepts,

and it lets us rapidly assemble new applications. It lets us visually construct the sequence of logic within our applications. It lets us compose new services from backend system connections, from Java components, from Web services. We can compose new services out of a set of existing services. It provides great consistency and the ability to reuse existing investments.

This technology, by the way, has received very positive analyst feedback. In the words of Ovum, for example, "This...reinforces IBM's position at the vanguard of the Web services movement; it demonstrates that in addition to promoting and collaborating in the development of Web services standards, IBM is capable of delivering real technology that people can use today" << said Gary Barnett, principal consultant with the Infrastructure Practice at Ovum >> "This is something that puts the company well ahead of many of its rivals."

And we would add to that, by the way, that this technology can help *you* get ahead of *your* competition as well. It offers you the next level in productivity, consistency, investment protection, and reuse.

Now let's look at slide 12. Here we consider the next step in the Credit Card process: Determine if the customer is "gold or above". If so, issue the card.

This requirement for "gold or above" is basically a business policy. You know, it's just the sort of thing that could change often. And in fact, there are potentially thousands of these kinds of policies, or rules, scattered throughout some customer applications.

What we'd really like to do is give our customers a way to create policies once that could be triggered in the source code, and then subsequently allow business analysts to maintain those policies without any additional programming required. In our case study example, maybe we'd like to issue a card to customers that are "silver or above", or maybe "platinum only". We might want to schedule when these kinds of policy changes would take affect. It's the sort of thing that could happen over and over again. By externalizing these kinds of variable business practices, we let business people run the business. The advantages are spelled out in more detail on slide 13. Let's take a look.

Slide 13 shows the benefits that come from externalizing business rules and policies.

We get explicit documentation of business practice decisions, clearer understanding of application behavior, reuse of policies across business processes, increased consistency of business practices, decreased maintenance and testing costs, improved manageability of business practice decisions, increased confidence in predicting the business impact of proposed changes, and the ability to easily identify and correct conflicting rules across the business.

All of this adds up to business agility, letting you adapt quickly to new business opportunity, well ahead of your competition.

Moving on to the third step in the Credit Card Application, which is repeated on slide 14, we see a requirement to check the customer's credit rating. We're going to do this by interacting with a Credit Bureau. You'll notice a couple of boxes in pink that illustrate IBM Web services leadership.

One of these is a private UDDI registry in which WBOnline can publish the set of tested and approved business services that its applications can use. A private UDDI registry is not actually required to make use of Web services, but it does deliver value in several ways. It facilitates Web services testing and development, and it provides a repository for service definitions that have been published for various applications to use.

You'll also notice a Web services gateway. The gateway facilitates configurable mappings between internally and externally defined services so that WBOnline can leverage the most appropriate implementation protocols based on the location from which a service is invoked. The gateway is fully extensible and can serve as a bi-directional control point for critical tasks such as security, validation, logging, transformation, metering, and so on.

To see one of the benefits from this gateway a little more clearly, let's consider a Web service invocation that's actually going from right to left (meaning that the Credit Bureau is going to invoke a service provided by WBOnline). We can assume that the Credit Bureau, like many businesses, might need to borrow money from time to time. So we could imagine the Credit Bureau invoking a Web service provided by WBOnline which provides a quotation of interest rates and repayment conditions for a given loan amount. Imagine that this very SAME service is used inside of WBOnline, in order to support a variety of *internal* applications. The Web services gateway allows this service to be implemented only once, and then accessed using the most efficient protocol that's available. So for example, this service could be used internally using an IIOP or JMS protocol, whereas the external definition of the same service would probably use something like HTTPS. The gateway would easily take care of the mappings between the two.

Moving on to slide 15, we can step back a little bit and reflect on the fact that steps one, two, and three in the Credit Card Application are actually part of a larger business process. Each step in this process is implemented as a service, and WebSphere Application Server V5 provides a robust environment for service choreography. We can choreograph services, not just in short running interactions like the CICS example discussed earlier, but also in longer running, stateful flows that involve both applications and people. The same underlying choreography engine is used in both cases. Please understand that the service choreography technology is literally *inside* of WebSphere, including access to full J2EE support. Remember that we had immediate access to Java-based connectivity back in step one. And when we got to step two, and we needed to determine whether the customer was classified as "gold or above", well then we were able to leverage WebSphere's J2EE-based business policy framework "in line". Indeed, we can take full advantage of Java business logic, and Enterprise JavaBeans, in the exact same operational environment in which our service choreography engine is executing. This has obvious performance advantages, it provides a rich and consistent environment for developers, it provides a common operational environment for administrators, and it's a great example of what I talked about earlier when I said that application servers are beginning to assume a much broader role in e-business solutions than they ever have in the past.

You know, one of the other interesting things about this slide is what happens when the Credit Bureau comes back with a response of "Customer Unknown". In this exceptional case, it's necessary to get people involved, in order to perform additional research manually, contact applicant references, and so on. WebSphere Application Server lets people as well as applications, components, and other systems participate in long running business processes. The people support that we're talking about here is not just limited to queries. WBOnline's staff, playing various roles, can participate in so-called interruptible flows which makes this technology very valuable.

Now let's see some additional valuable extensions that are described by slide 16.

On slide 16, we see what happens when a credit card is actually issued. There are interactions with remote systems using messaging technology (that's really referring to the systems that handle the physical production of the credit card and the associated mailing process), and there are also interactions with CICS using the External Call Interface (or ECI). That's where the credit card information gets stored in WBOnline's business system.

In the case of the messaging technology, note that the remote system interactions are hidden behind a component interface, represented by the top gray circle. WebSphere makes it possible for J2EE developers to interact with inbound and outbound messages in a transparent fashion. They can view the world as a set of components in which the details of the Java Message Service (or JMS) are effectively hidden. We describe this at the top of the slide as advanced messaging. You'll also see us talk about advanced transactions. Now what do we mean by that? Well, in that case we are talking about the larger orange circle that's shown in the figure. There we are indicating that both the messaging update, and the CICS update, need to be treated like a single unit of work. We want either both of the updates to occur, or neither of the updates to occur. Now messaging environments like MQSeries are frequently used in a distributed transaction, but lots of environments simply don't support this. And in fact, it's not possible to access CICS through ECI as part of a distributed transaction. In spite of this, WebSphere *extends* the transactional environment so that a resource which normally doesn't participate in a distributed transaction is enabled to do so. That's another significant innovation from IBM that can bring real value to your business.

The final point being made at the top of this slide is that WebSphere supports advanced component models. One example of this is indicated by what we call "Advanced Query Services". WebSphere makes it possible for WBOnline to order search results, to include date and time expressions in queries, and to literally run dynamic queries in which the details about the information being gathered are not fully understood until the application is actually executing. As one consequence of this dynamic query support, it's possible to reduce the amount of code that needs to be written. It's also possible to obtain the information that's needed in certain application scenarios much more efficiently, in fact, the processing can be more efficient by an order of magnitude as compared to other vendor's application servers.

So, we've really said a lot here. We are talking about a number of things that are not yet addressed by J2EE. And we are talking about things that are typical of what many businesses are currently trying to do. Now let's move on to slide 17.

On slide 17 we introduce another WBOnline application. This one is responsible for year-end tax updates.

Think for a moment about how this type of application would run in a J2EE environment. You'd want to schedule when this kind of application would run. It should be "unattended", in the sense that it doesn't require any human-computer interaction. Instead, it would run in the background, and probably for a long time. It might be broken into several parts that would run in parallel. For those of you that have been in data processing environments for a while, we would call this sort of thing a "batch job".

Please don't be mislead by the term "asynchronous" on this chart...this is not about MQ or messaging-based processing. It's about processing that doesn't have someone waiting on the other end for a specific response.

Why is this kind of application interesting to you? Because this is not easy to accomplish with traditional J2EE application servers. And yet this kind of processing requirement is fairly

common, and WebSphere Application Server *lets* you do this. Let's keep moving...it's time for slide 18.

Now we consider a travel application. And we are still focused on extensions at this point. The optimizations discussion will be coming up soon.

The business scenario has now evolved to incorporate a Travel Agency as another business partner. The Travel Agency advertises trips on the WBOnline Web site. Customers who book trips represent a valuable revenue source to the Travel Agency. WBOnline benefits from this relationship because the trips are paid for through a loan that's extended to the customer.

What are the steps involved in booking a typical trip? As shown on this slide, you'd need to book a hotel, book a car, and book a flight. Suppose that the first two parts of this transaction succeed, but the third one fails. What is the implication? You'd need to somehow reverse the first two transactions. You'd need to "undo" them, or compensate, for the work that was already committed.

WebSphere automates this scenario by providing *automatic* "undo", or compensation processing. As a developer, you only need to register the compensating step, using graphical tooling, and the runtime environment will then automatically take care of executing the compensating step when needed. This saves programmers from the task of writing extra, potentially error-prone, code.

At the bottom of the slide we illustrate that the precise nature of compensation can be arbitrary. Once you've taken a bite out of an apple, you really can't put it back. But you *can* write someone a promisory note, indicating that you'll replace the apple at some point down the road. The point here is that WebSphere can drive compensation, in whatever form that appears, in an automated fashion. That means less time for rolling out complex applications, and with higher quality. Let's look at slide 19.

On slide 19, we are reminded that the WBOnline applications we've discussed need to be delivered to international constituencies. A smart server needs to account for different cultural conventions, languages, and geographical considerations. It should also be "time zone aware" with respect to clients. WebSphere Application Server meets this requirement through an internationalization service that lets you properly handle date and time information, formatting of currency and decimal points, language selection, sorting rules, and more. Please note that these capabilities are not just presentation-related. They are very relevant to business logic as well. WebSphere applications can easily leverage internationalization context across a distributed application end-to-end. This lets your business applications adapt quickly to new geographical constituencies. That means increased flexibility and productivity for your business.

On slide 20, we see other valuable extensions. We can talk about Shared Work Areas: Efficiently sharing arbitrary information (e.g., customer profiles) across a distributed application using "virtual scratch pads". ActiveX Bridge: Invocation of EJBs from Microsoft clients and servers with high qualities of service. CORBA Integration: Bi-directional communication between popular third-party ORBs and J2EE applications (with documented restrictions). CORBA C++ SDK: Integrating C++ assets (both clients and servers) into J2EE environments using IBM's CORBA Software Development Kit.

Well that's a big plate full of information, but we've got some really good things that are still ahead of us. I hope that your seat belts are fastened, because it's time to look at slide 21...this might just be the best thing we've got in the basket.

Slide 21 lets us switch gears to talk about optimizations. Recall that what we've discussed so far in the case study is about extensions...things that affect the source code that is written or generated. Things that extend the programming model, and that are commonly associated with development tools. Now we change over to optimizations, which do NOT affect source code. These are the kinds of things that are deployment-related and administration-based, and they relate to greatly enhanced performance, availability, and scale.

In the context of our case study, let's assume for a moment that WBOnline has made the decision to store customer account information in a DB2 database. Let's say that they have also made a decision to make appropriate use of Enterprise JavaBeans, and that one of the business objects they've created is an Account component. This component provides services to the Credit Card Application, the Year-End Tax Application, and many others.

Now the *way* in which these applications need to use this common component is going to vary greatly. For the Credit Card Application, account information for a particular customer is retrieved but not updated. Indeed, the update occurs elsewhere in the application - to a CreditCard component, as you would expect. Since this application is "user facing", fast response time is required.

The Year-End Tax Application, on the other hand, runs as a background task. As such, it can execute with a lower priority. Moreover, the nature of the application is more about update than retrieval, and the scope of this application extends to all customers, not just one customer.

To accommodate these diverse computing requirements, WebSphere Application Server provides profile-based access to data sources (this is also known as application profiles). What this means is that we have a powerful mechanism for instructing the same component (or more accurately, the so-called container or environment surrounding that component) to interact with a database; a workload manager; or any other part of the runtime infrastructure in a fashion that best serves the needs of the application which is currently running. Do you need to access data in a read-only fashion? Do you want to prefetch data before it's needed? The source code of the component should not be affected by the answers to these kinds of questions, and the need to redeploy the component should be minimized. The net effect is this, and this is the key point, with application profiles the very same component can provide optimized services to different types of applications at exactly the same time as a result of administrative decisions. Now that's mind-boggling. This leads to unparalled opportunities for reuse, *without* sacrificing runtime efficiency. This technology is a unique and valuable innovation delivered by WebSphere's product engineers.

Now let's look at more of the optimizations story, as told on slide 22.

On slide 22, we see additional optimizations (some of which are scheduled for delivery later this year), and these things speed database access; promote high availability and efficient utilization of resources; support pluggability of third-party and custom built software; and allow the runtime environment to adapt itself to actual usage patterns.

You'll notice that WebSphere optimizations extend right out to the "Edge of the Network". This provides WBOnline with additional performance, availability, and scaling advantages in highly dynamic and distributed configurations.

With WebSphere, you get end to end intelligent load balancing that works with existing appliances, as well as advanced caching services.

WebSphere also provides content distribution services for secure deployment of published Web content to caches and re-hosting servers throughout a distributed network. This pre-positions content closer to the user for an enhanced user experience.

You get policy-based quality of service which means that computing and network resource allocations are based on a transaction's real-time business value. This prioritizes networking throughput for the most highly valued customers.

And then of course there are security services...these include support for the Secure Sockets Layer (or SSL) protocol, together with SSL endpoint affinity and optional hardware-assisted cryptography for enhanced and secure connections to Web resources. You can also integrate Tivoli Policy Director to achieve a centralized security solution that includes user authentication at the edge, cross-domain single sign-on, and web-based access control for blocking unauthorized access of both cached and non-cached Web resources.

Slide 23 takes the story even further...

It's hard to talk about optimizations without mentioning z/OS. The z/OS platform delivers innovations that you'll find nowhere else, and WebSphere on z/OS leverages these things natively. We're talking about the ability to handle unpredictable spikes in workload, with even and consistent end user response times based on business policy. We're talking about the ability to recover transparently from a major subsystem failure such as DB2.

For the very highest qualities of service and the most demanding processing requirements, look for the strengths of z/OS that are truly unique to an IBM solution.

Slide 24 brings us to the point where we are ready to map what we've heard to a variety of WebSphere foundation and tool offerings, so let's continue straight ahead to slide 25 without delay...

Slide 25 is a reminder that when we talk about WebSphere Application Server, we are looking at a core application server that supports J2EE and Web services.

A set of extensions (shown at the top) deliver additional productivity, flexibility, and integration capabilities -- exactly as we described in our case study. You should really think of these extensions as delivering "tomorrow's standards today". We are actively bringing several of these capabilities into J2EE as part of the Java Community Process. There are formal Java Specification Requests that have been submitted, or that are planned, for many of the features that we've discussed. And we are also active with the W3C, and other major companies in this industry, to drive IBM innovations into an approved set of open standards.

Now looking at the bottom of this slide, you are reminded that WebSphere also includes a set of optimizations that enhance the application server's performance, availability, and scaling characteristics. You'll notice that there's a subset of the optimizations shown in a pink color. The idea here is that not all optimizations are created equal. At one level we can talk about things like failure bypass and load balancing. But there's a second level of enterprise optimizations that raise the bar even higher by delivering things like cross-domain failure bypass and dynamic load balancing in support of enterprise-class deployments.

Keep this figure in mind, as we move on to slide 26 where these concepts will get mapped to a number of application server offerings.

On slide 26, we see several configurations of WebSphere Application Server on distributed systems. You should really think about the application server in terms of three different products. For customers who want all that WebSphere has to offer, there's WebSphere Application Server Enterprise which is a follow-on to the Enterprise Edition. The Enterprise product offers the best in extensions and optimizations, and it even includes an MQSeries license for enhanced integration support. With the Enterprise offering, there's a lot of flexibility in terms of what you choose to deploy...for example, you could deploy software for business policy management on a single stand-alone server. Or, you could deploy the full set of extensions onto a large network of machines. The choice is yours. Notice that this product has nothing to do with TXSeries which is being broken out as a separate product. You might also be aware that Enterprise no longer ships the Component Broker runtime. We've re-engineered the high end runtime around a common code base, which is the strategy that we've had in place for quite some time now. We do have some special CORBA capability in the high end product, but it's an extremely small subset of what the offering is really about.

Looking down at the bottom of this slide, you'll see another product called WebSphere Application Server Express. This is the follow-on to the Standard Edition. As you can see, it provides a J2EE and Web services subset. It includes support for JavaScript and tag libraries, as well as the ability to consume Web services. There's no support for EJBs, and in fact you won't need to be a Java programmer to use this offering. Instead, there's a focus on ease of use, small footprint, pre-canned applications for high productivity, and integrated tool support...indeed, the WebSphere Studio Site Developer code is provided with the runtime in a single integrated offering.

Then you can look at the middle of the chart. Here's where we see WebSphere Application Server, which offers three different deployment options. From bottom to top, there's the single server environment. This is a follow-on to today's single server configuration of Advanced Edition, with the latest standards support for J2EE and Web services, as well as JMX (or Java Management Extensions) support. So this would give you J2EE 1.3, which means things like EJB 2.0, an embedded JMS provider, support for message-driven beans, support for J2EE Connector Architecture, and so on. Then, we get optimizations in the form of things like clustering support, and that's the Network Deployment option. This would be the follow-on to the clustered Advanced Edition offering. What are some key enhancements here? Well, we've actually made a lot of improvements to the Advanced Edition. This includes a simplified browser and XML-based administration model, the ability to add single-server configurations of WebSphere Application Server into a clustered environment with ease, in-memory replication of HTTP session state and cached page fragments, increased amounts of caching with increased resilience in the face of component or system failure, removal of the requirement for a relational database as a configuration repository, improvements in workload management to allow weightings in the selection of servers, simplified problem determination with first failure data capture, automated collection of logs and so on when a remote services team needs to get involved in trouble-shooting, extensive use of JMX for systems management and performance monitoring with architected pluggability to third-party tools, security enhancements related to JAAS, CSIv2, and Java2, and much more. Of course, full support for J2EE 1.3 and the latest support for Web services standards goes without saying -- remember these things are like Russian dolls. Any improvement at the single-server level is automatically enjoyed at the clustered level. And finally, we see the high-end deployment scenario played out at the top, with

an Extended Deployment option which is initially only in beta. There's really no precursor to Extended Deployment. You can see this because the little pink box on the right is blank. This option is being introduced for those customers who need IBM's most optimized environment but who might not yet be ready to use the extensions. This is where you get into things like SQLJ for database access, the powerful application profiling support that I described earlier, dynamic workload management and pluggable workload management, cross-domain failure bypass, edge-of-network optimizations (ultimately) for content distribution, policy-based quality of service at the edge of the network, and things of that nature. This is also where you'll find asynchronous binary distribution to facilitate application deployments where one or more nodes in the network might be down.

Now, at this point you might be wondering how this slide relates back to the case study discussion. How are the things in the case study addressed by the Enterprise offering, for example, as compared to the Network Deployment offering (which sits right in the middle of this chart)? The next slide takes a look at that...

On slide 27, we see the functions that were referenced in the case study. As shown, the Enterprise offering ships with everything that's needed to support WBOnline's requirements. The Network Deployment configuration (which is a follow-on to the previous clustered Advanced Edition offering), well that addresses some of the WBOnline requirements, but in most cases it's going to be the Enterprise offering that gets the job done. Now this is not to say that the Network Deployment offering isn't valuable. In fact, we've done a lot to enhance what you think of as today's Advanced Edition. This includes a simplified browser and XML-based administration model, the ability to add single-server configurations of WebSphere Application Server into a clustered environment with ease, in-memory replication of HTTP session state and cached page fragments, increased amounts of caching with increased resilience in the face of component or system failure, removal of the requirement for a relational database as a configuration repository, improvements in workload management to allow weightings in the selection of servers, simplified problem determination with first failure data capture, automated collection of logs and so on when a remote services team needs to get involved in trouble-shooting, extensive use of JMX for systems management and performance monitoring with architected pluggability to third-party tools, security enhancements related to JAAS, CSIv2, and Java2, and much more. Of course, full support for J2EE 1.3 and the latest support for Web services standards almost goes without saying. But the point still stands...there are going to be harder problems than the Network Deployment option can solve, and that's exactly why we have an Enterprise offering...to handle the kinds of realistic challenges that WBOnline faces...which J2EE and Web services are not yet able to address.

So that summarizes the foundation offerings...now what about the tool offerings? This naturally brings us into a discussion about WebSphere Studio, which is covered on slide 28.

Slide 28 shows that WebSphere Studio is IBM's single, comprehensive integrated development environment. It is an extremely flexible, extensible, and pluggable environment, and it's based on the so-called Eclipse technology that's now part of the open source community. This same technology forms the basis of the WebSphere Studio Workbench, depicted at the bottom of the figure. You can really think of this fully-supported workbench as an "operating system for tools", in which common tool-related functions are provided (such as user interface services, desktop and help frameworks, project management and debugging services, and common

file-based source control management services). Into this single workbench infrastructure IBM plugs various WebSphere Studio tools, which provide complete *award-winning* support for all members of the WebSphere Application Server family -- award-winning as in the most valuable product award at JavaOne, for example. Also shown on the figure you'll see that additional IBM tools provide support for other IBM middleware. And, as shown, there are partner tools that complete the story with additional functionality covering all aspects of the software development life cycle. Note that more detail about some of WebSphere Studio tools is provided on the next slide.

Looking at slide 29, you'll see various WebSphere Studio configurations that are packaged to address different developer roles and requirements. These include capabilities for Web and Advanced Web development, J2EE development (that's handled by WebSphere Studio Application Developer), extensions to J2EE such as those required by WBOnline (again, you'll notice that's covered by WebSphere Studio Application Developer Integration Edition), and then the WebSphere Studio Enterprise Developer completes the picture at the top by adding fourth generation language technology and support for things like COBOL running on mainframe systems.
Regardless of your requirement, WebSphere Studio tools can address your development needs through a single, well-integrated development environment. Slide 30 illustrates additional advantages that WebSphere Studio users will enjoy.

Slide 30 illustrates that, regardless of the runtime that's chosen, IBM provides associated tools that help maximize developer productivity. For the Express offering, the tool and runtime are already integrated into a single package. For WebSphere Application Server, we recommend either WebSphere Studio Site Developer, or Application Developer, depending on a customer's needs and the roles that apply to the customer projects. For the Enterprise runtime, we recommend Application Developer Integration Edition. The Integration Edition actually has the Enterprise runtime embedded into it as the unit test environment. And for z/OS, we recommend Enterprise Developer, or whatever might be the most appropriate depending on how sophisticated the development effort turns out to be.
It's critical to note that the integration of IBM's tools and runtimes are seamless. The environment that you use for development can provide an equivalent programming model to the environment that you use for deployment. This promotes efficiency and consistency in the creation and maintenance of high quality business applications. In the words of Tim Hilgenberg, Chief Technology Strategist for Hewitt Associates, "With WebSphere, we win both ways: we get the flexibility and robustness we need on the *production* side, along with full support for all our current *development* activities - from Web pages to server components to databases." << end of quote >>
We would suggest that it's the deep *integration*, integration between the runtime and tools that provides you with such a significant advantage when choosing IBM. Now speaking of integration, let's look more closely at the idea of integration as it relates back to the WebSphere triangle graphic that we used in the first part of this session.

On slide 31, we return once more to our WebSphere triangle with some positioning around the *general* notion of integration. The point of this slide is to provide a mapping of different integration styles to the overall WebSphere portfolio. This helps to ensure that you choose the

right kind of WebSphere offering based on the kind of problem that you are trying to solve. We actually need to go beyond the foundation and tools dimension in order to have this discussion. Let's start in the upper left hand corner of the triangle (empower users with personalized and aggregated information). Some customers need an integrated user interface for accessing diverse applications and content. They are looking for a rich user experience characterized by personalized and streamlined access to both content and collaborative services. Key objectives in this case include support for pervasive devices, single user signon, pluggability and extensibility in order to efficiently support new portal sources, powerful and flexible real-time search capabilities, and analytics support for measuring portal effectiveness.The IBM offering for this integration requirement is the WebSphere Portal solution.

Moving to the right, we see some words about automating and optimizing business processes. Some customers are looking for the automation of business processes that span complex, heterogeneous, legacy environments - these environments frequently consist of packaged applications such as SAP and Siebel. They may also require business process integration of people and organizations. Key objectives in this case include the modeling, managing, and monitoring of both intra- and inter-enterprise business processes in order to promote business agility and to achieve high return on investment; support for robust connectivity; support for business-oriented cost analysis; and rapid deployment achieved through process templates. The IBM solution for this integration requirement includes IBM CrossWorlds offerings, MQSeries Workflow, and complementary offerings from Holosofx.

Moving to the lower right corner, we see the need to unify information for a "single view" of key business data. Some customers need to unify, connect, and integrate diverse information for a single view of key business data. This includes the loose coupling of heterogeneous application information and data via transforms, routing, and messaging. It also includes services that provide message enrichment and data capture associated with message flows. Key objectives in this case include dynamic distribution and streamlined routing of information between different application sources, rules-based information connection, and content-based publication and subscription services. The IBM solution for this integration requirement includes both WebSphere MQ and WebSphere MQ Integrator.

Finally, we see the requirement to rapidly assemble reusable solutions leveraging existing resources. Some customers, and this applies to WBOnline, are strongly committed to J2EE and to the development of new Java-based applications that will be deployed into an application server environment. These new applications frequently need to leverage existing legacy assets in combination with the development of new business logic written in Java. Key objectives in this case include high operational productivity through a single administrative environment, high development productivity through an integrated development environment, standards-based connectivity, strong support for component models, and flexible "service choreography" facilities for defining both fine and course-grained flows (including people interactions). The IBM solution for this integration requirement includes WebSphere Application Server and WebSphere Studio.

Now there are a couple of critical things that you should note about all four solutions:

These solutions are not mutually exclusive. For example, customers using CrossWorlds and who need new J2EE applications with fine-grained integration to existing assets will also want to use WebSphere Application Server. Customers using WebSphere Application Server and who need support for information transformation and routing will also want to use WebSphere MQ Integrator. Customers using WebSphere Portal Server for integration "on the glass" and who also

need to achieve integration between diverse backend systems will also want to use one or more of the other IBM integration solutions.

Taken together, these solutions represent the industry's most complete integration story. IBM is exceptionally well positioned to address the broad and diverse requirements that are found in the "integration space".

Slide 32 steps back for a moment to consider the value of WebSphere Application Server as part of an entire platform. We think that there are a lot of good reasons to choose WebSphere, and we've listed some of those here...

First of all we can talk about support for complete solutions. The value proposition for WebSphere Application Server starts with support for complete solutions. WebSphere Application Server provides the foundation for a complete e-business infrastructure that spans everything from "reach and end-user experience" to deep "business integration". When WebSphere won Network Computing's Editor's Choice award, the magazine stated that "WebSphere Application Server is part of IBM's larger suite of complementary products that sits on top of this world-class solution." Network Computing's most compelling remark about the application server was actually based on the things that run on top of it - the full set of solutions that it supports. Of course, WebSphere also benefits from IBM's expertise and ability to supply end-to-end services and enablement; the complementary experience and capability delivered by the Lotus, Tivoli, and DB2 brands; clear hardware leadership; strengths in Global Financing; and a set of very strong industry partnerships. And this discussion about complete solutions would not be complete if we didn't also mention integrated tools - one of IBM's key differentiators.

We can also talk about unparalleled sustained investments. WebSphere Application Server is backed by IBM's year-over-year multi-billion dollar investments and many thousands of IBM employees, with industry-leading developer programs that promote widespread WebSphere adoption. WebSphere has been one of IBM's "must-win" initiatives with top-level executive commitment and extensive focus across the company. WebSphere has also benefited from IBM's enormous investments in research and development (some $5.6 billion dollars in 2001 alone). There were 524 patents attributed to IBM software this past year - more than any other software company in the world. Of these 524 patents, 222 were specifically related to WebSphere .

What about core strengths and competencies? WebSphere Application Server incorporates IBM's core capabilities in building system software. These capabilities include transactional and security leadership; an ongoing focus on interoperability; IBM's heritage in delivering messaging and persistence capability as well as component technology; strengths related to Web services and XML; industry-best support for manageability (including synergies with Tivoli); and significant experience in the area of application integration and connectivity.

And then we could look at clear industry leadership. WebSphere Application Server has very strong industry momentum that is second to none (as measured by both analyst and press opinions, market share numbers and trends, and growing developer momentum). WebSphere has won key competitive battles in accounts like eBay and Abbey National, and more than 50,000 customers are now using WebSphere worldwide. Business partners are also increasingly selecting WebSphere as their application server of choice. WebSphere delivers first-rate support 24 hours a day, seven days a week. WebSphere Application Server and Java solutions from IBM have together won more than 30 industry awards. Beyond garnering an impressive number of industry awards, WebSphere has allowed customers such as Bekins Van Lines to receive first-place industry awards of their own.

We would suggest that WebSphere provides excellent return on investment. WebSphere Application Server helps businesses improve customer loyalty, respond more quickly to change, and reduce overall costs. This last point hints at WebSphere's advantages related to overall cost of ownership measured in terms of truly excellent price/performance, as well as the provision for customers to leverage their existing investments. WebSphere customers enjoy an integrated solution stack where each component has been engineered and tested to work within a larger solution space. Again, this solution stack includes integrated tools - tools that make it possible to build and deploy high quality e-business applications faster than ever before.

What about support for industry standards? WebSphere Application Server is J2EE-certified on more platforms than any other vendor. WebSphere is the first and only major application server brand to become J2EE 1.3-certified last year (through the Technology for Developers release). WebSphere engineers have contributed to more than eighty percent of the J2EE specification, and these engineers continue to define the next wave of standards through participation in the Java Community Process. WebSphere was first in line to deliver an integrated Web services solution with a complete set of associated tooling. WebSphere engineers have defined, co-authored, or significantly contributed towards all of the relevant standards in the Web services and XML space.

And last but not least, we can talk about scale and performance across platforms. WebSphere Application Server scored first place in PC Magazine's public benchmark. WebSphere has also delivered superior price-performance in realistic configurations as part of the public ECPerf benchmark. Beyond that, WebSphere demonstrated the scale and performance results necessary to secure a victory in the highly visible eBay bakeoff. According to Evan Quinn, chief analyst with the Hurwitz group, the eBay contract suggests that WebSphere is "reliable enough to handle the huge transaction volume served up by even the busiest Internet sites". WebSphere customers are able to achieve many tens of millions of page views per day in production on the Web. Note that Kana set an eCRM Solution industry benchmark by running on WebSphere. And IBM has demonstrated more than 12,500 EJB-based transactions per second - serving 800,000 users - with approximately one quarter of a second average response time. WebSphere delivers the industry's best vertical scaling, and only WebSphere leverages the native scaling facilities that are offered on z/OS. WebSphere enhances base application server performance through advanced caching technologies and edge of network performance enablers. WebSphere is supported by a High Volume Web Site team, and by a High Volume Web Site Performance Simulator used in capacity planning. WebSphere includes performance-monitoring APIs and tooling that lets customers measure application load characteristics with both IBM and third-party software. WebSphere offers performance wizards and is backed by substantial performance-related collateral (such as redbooks and white papers).

Taken together, we believe that all of these things spell SUCCESS for your business.

Slide 33 illustrates the value of WebSphere as recognized by UBS, one of the world's largest banks. For your business, just as for UBS, WebSphere can provide a secure and robust environment that helps you build a loyal customer base. The integration features shorten development cycles, reduce costs, and protect your existing investments.

Now we're ready to talk about some next steps that might be considered, so we'll move right beyond slide 34 and spend some time reviewing slide 35.

Slide 35 points out that WebSphere is not just about software. It includes the idea of services and education, in order to help you be as successful as possible in applying this technology. WebSphere services are available for transferring deep technical and product skills to your organization, with a focus on best practices. WebSphere services mean that you don't need to be afraid of putting the latest technologies to work. From the front-end of the software lifecycle, with a focus on architecture and design, to ongoing application development efforts, to the monitoring and tuning of deployed systems, you can count on WebSphere services to deliver everything you'll need to accelerate the completion of successful WebSphere projects.

You'll see on this slide that WebSphere education is available as well...including both public and private courses, with roles-based customized education, and either onsite, or online training, as part of a distance-learning curriculum.

Now you should be aware that there's a very special services offering related to WebSphere that deserves extra attention...and that's shown on slide 36.

Of particular interest to many customers is the fact that IBM has developed some special "Center of Excellence" services related to WebSphere. A Center of Excellence results from an enterprise commitment to improve software development, delivery, and operations. It typically provides:

Technical and managerial know-how

A shared repository of technical information, best practices, and corporate standards.

Repeatable processes supported by reusable assets, and

Accelerators for proliferation of technology throughout the organization

So why would you want to invest in a Center of Excellence? Simply to increase the successful adoption of new technologies, to improve return on investment, and to gain significant competitive advantage. These things result from a combination of best practice documentation; monitoring and performance tuning strategies; testing strategies; architectural standards and design patterns; well defined roles and responsibilities; time-saving code samples; repeatable processes; skills enablement; problem determination, isolation, and management; backup and recovery procedures; and more.

Notice that IBM makes these services available in flexible suites. An overview suite gets you started with best practice guidelines. A mini suite is designed for a business unit with a single application focus, and it adds infrastructure and deployment readiness reviews; a focus on high performance; and mentoring services related to application development and infrastructure. For the most comprehensive enablement, consider the full suite for enterprise-wide e-business competency and expertise with a laser-sharp focus on process, on skills, and on organizational structure; roles; and responsibilities.

At last, we come to slide 37. This is basically a reminder that WebSphere is ready to make e-business your business!

WebSphere delivers innovative J2EE extensions that promote integration, flexibility, and productivity. Think about things like the integration capabilities of Enterprise with support for service-oriented architecture, the flexibility to run dynamic queries, and to externally administer business policy. Also consider the productivity benefits from the frameworks and tools that are delivered over a range of offerings...from Express up to Enterprise.

WebSphere also delivers industry-leading optimizations that boost performance, availability, and scale. We talked about things like application profile support, dynamic workload management, and integrated edge of network technologies.

These capabilities are packaged in a set of flexible configurations that address different business needs. Taken together, optimizations and extensions help you cut costs, boost customer loyalty, and adapt quickly to change.

But you should also remember that WebSphere extends beyond application serving. WebSphere is a complete platform with integrated tools and value-added products that maximize reach and end-user experience along with strong support for deep business integration.

WebSphere is also complemented by services and education, as well as a world-class Center of Excellence program designed to help you speed the successful adoption of new technologies.

WebSphere is more than ready to help you enjoy competitive e-business benefits in record time.

Thank-you *very* much for your attention, and have a very good day.