1 Overview of IBM Device Strategy

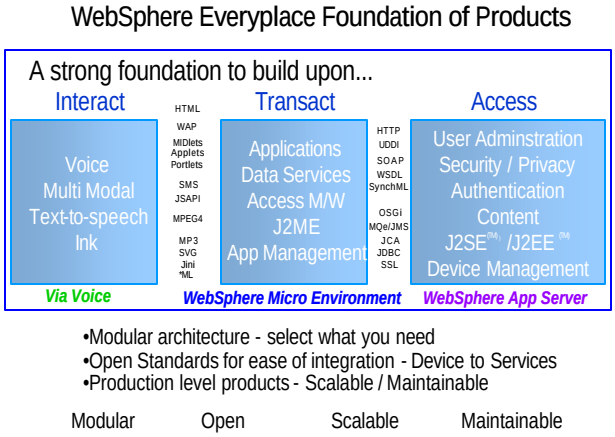
There is an overwhelming emergence of small, convenient devices that combine cellular communications with compelling content. In Hand devices such as cellular telephones, smart phones, Personal Digital Assistants and hand held computers all have games, entertainment and personal productivity applications such as Personal Information Management (PIM) Applications including contacts, calendars and to-do lists.

The next generation of data services is at hand. Creating demand for new high-value data services to drive up Average Revenue Per Unit (ARPU) on connected devices is the major objective of every carrier and service provider. To provide these new high-value data services, carriers and Original Device Manufacturers (ODMs) are focussed on the following three areas. How the device *interacts* with the user. How the device can be used as to *transact* in a business sense. Finally, how the device can *connect* to other systems.

1.1 Higher value data services equals higher value devices

IBM's device strategy is to use open industry standards to combine the power of WebSphere® and the portability of J2ME™ Java™ technology with the convenience of cellular telephones, smart phones, PDAs. By enabling these devices to interact, transact and connect to existing IT applications running major businesses today, IBM. This will allow the creation of higher-value data services for carriers to host, provision and maintain to an increasingly discriminating subscriber customer set. By enabling these higher-value data services, ODMs can also differentiate their devices to customers (Carriers, Service Providers, OEMs and Enterprise IT).

To address the need for higher-value data services, IBM has created the WebSphere Everyplace Foundation. This architected collection of integrated products allows service providers to host new compelling data services that enhance the end-user interaction, transact with business systems and connect to existing applications running in e-businesses all over the world; driving increased connection and increased downloading of new data services. Figure 1.1 illustrates the Foundation.



1.2 Figure: WebSphere Everyplace Foundation

By combining the power of WebSphere, with the portability of Java technology, IBM delivers the convenience of low cost, mobile and pervasive devices to e-business.

Through the integration of software from device platforms to middleware and through to the server and server applications over varying types and capabilities of wireless networks, IBM has delivered a platform where carriers and service providers can easily create, deploy and manage this new generation of data services. IBM provides a tested and proven middleware offering allowing carriers and service providers to focus on applications that differentiate their services and for their customers and improve the user experience.

The Foundation is not monolithic. The architecture of the products is modular in nature. The power of open standards is revealed. Not all components need be selected. Vendors can select what they need and are free to substitute alternate technologies as they see fit. Since the Foundation is based on open and industry standards, other systems can also be integrated quite quickly. Several IBM business associates have already produced value-added components such as Bluetooth, MPEG encoders / decoders, Relational Database, Jini, Secure Socket Layer (SSL) and many other solutions tested on the Foundation. The architecture is also scalable and extensible and allowing for quick addition of extensions for individual carriers.

1.3 Interact

As applications are extended to mobile devices, new technologies will be required to provide productivity to the end-user while shielding them from the inherent complexities of integrating local and remote services. Local services such as barcode scanning, displaying scalable vector graphics running full-motion video and voice navigation will be required on the device. Remote services, residing within the service infrastructure, such as location (GPS), transcoding and messaging services will need to be accessed wirelessly by the device. The merging of local services with remote services allows complete flexibility for carrier running the full range from 'always on' to 'available for download'.

By allowing human machine interaction (HMI) using various types of inputs such as speech recognition and text-to-speech, IBM ***Embedded ViaVoice®*** enables a new, rich multi-modal end-user experience. Using device-based or server-based speech engines, voice navigation, command and control or text-to-speech technologies, devices can become an extension of the user, even if the user has no direct access to a browser or other application.

Name recognition allows for hands free dialing. Dialers and command control features allow users to navigate through device and application functions using voice commands. IBM's Via Voice solutions are sensitive to the processing power and memory limitations of in hand devices and offer a full range of on device, and device to server voice functions.

For multi-media interaction with an end user, various technologies can also be used. Through the use of streaming video, MPEG4, and scalable vector graphics (SVG) images can be shown on small devices.

1.4 Transact

IBM ***WebSphere Micro Environment*** provides the underpinning platform for the deployment of e-business applications to small mobile devices. This is known, in e-business terms, as infrastructure.

Supporting a variety of processors (ARM, StrongArm, Xscale, MIPS, PowerPC and others), and popular operating environments (OSE, BREW, Linux, Rex, QNX, PocketPC, and PalmOS to name a few), WebSphere Micro Environment contains a production-ready Java Powered™ runtime environment for all in hand devices; and a whole lot more. This environment has been tested and certified to meet the J2ME™ specifications (CLDC, MIDP, CDC, and Foundation) as specified by the Java Community Process℠.

IBM's value is to combine certified J2ME compatibility with middleware and data integration to provide a complete platform for extending e-business applications onto millions of devices connecting to backend systems worldwide. A complete description of J2ME can be found in appendix I.

Middleware allows the device to seamlessly manage the challenges of intermittent connectivity that occur in today wireless world. A handheld device, taken out of range, or a PDA taken deep into a warehouse while taking inventory, are no longer rendered inoperable. Data can be stored locally on the device in using DB2e® or other JDBC compliant relational databases. The data can be synchronized, as a connection becomes available using JMS or MQe. The use of assured messaging lets service providers explore the application possibilities for semi-connected devices despite the connectivity challenge without the need for user intervention.

By combining data, transactions, applications, and middleware, WebSphere Micro Environment delivers a platform for applications ranging from Personal Information Management (contact lists, schedules, browsing, SMS) to extensions of corporate IT applications and beyond to B2C e-business transactions. It also expands PIM applications to link and to extend existing e-business applications (data, transactions and applications) onto wireless devices, creating a new class of applications known as Enterprise Information Management (EIM).

1.5 Connect

As new higher-value data services are created, they need a server on which they can be hosted, maintained, and controlled. IBM ***WebSphere Application Server*** offers a world-class infrastructure for the next chapter in open e-business platforms. As the foundation of the WebSphere software platform, WebSphere Application Server provides a rich, e-business application deployment environment with a complete set of application services, including capabilities for transaction management, security, clustering, performance, availability, connectivity, and scalability. Improve time-to-value by building new integration-ready applications that leverage existing enterprise data services. Services such as access to relational databases, transactions, ERP systems and CRM systems, as well as, B2B and B2C solutions. WebSphere Application Server provides a scalable infrastructure to meet the growing workload generated by Web-enabled applications. New implementations of open standards for J2EE and for Web Services combine to create a powerful integrating platform and an "OS-like" network-aware foundation offering profound improvements in productivity.

1.6 Manage

Carriers and service providers understand the seriousness of putting an application in production. The application will be deployed to millions of devices, and will need to be updated on a regular periodic basis. Subscribers will also want the ability to add, discontinue or modify their service offerings, without the need for a visit to the carrier or service center. Applications need to be provisioned and maintained over-the-air (OTA or OAP).

1.7 Tools

Tools are a very important part of the solution for the development, testing, deployment and maintenance of total solutions. IBM brings consistency across many tools to leverage the skills of programmers across the typical chasms of device and server application development. Bringing the Internet programming to these solutions allowing programmers to focus on solutions rather than bits and bytes.

IBM has introduced WebSphere Studio Device Developer, the evolution of its award winning Visual Age Micro Edition embedded Java Development Environment, by building on top of the widely supported Eclipse open tools platform. The eclipse platform provides a plug-in architecture and serves as the platform for other IBM product including WebSphere Application Developer. WebSphere Studio Device Developer provides developers with a complete build, deploy and test environment for the creation of Java applications that target WebSphere Micro Environment. As part of the
WebSphere Studio family of products, it allows developers to build, test, deploy and maintain end-to-end applications. These applications can communicate and interact with J2EE applications developed using its sister product WebSphere Application Developer, which is also built upon IBM's WebSphere Studio Workbench.

1.8 Putting it all together

End-to-end e-business does not happen in a vacuum. Existing enterprise databases can provide millions of users with up to the minute information, on demand or transparently based on current business needs. Existing banking applications provide financial services through thousands of ATMs simultaneously. More than a stand-alone virtual machine, it takes integrated severs, middleware, device software to create end-to-end e-business solutions. Supporting the foundation it also takes education, business associates, system integrators and a development community. By extending enterprise applications to cellular handsets and smart phones, carriers have a means of accessing enterprise data, transactions and applications to offer new, more sophisticated Data Services to their corporate and professional subscribers. Information Technology (IT) also has a means of extending existing e-business applications to mobile, wireless and pervasive devices.

IBM WebSphere Everyplace Foundation provides the platform for the deployment of e-business applications to small mobile devices. By using open standards to combine the portability of Java™ technology with the power of WebSphere, IBM delivers the convenience of mobile devices to e-business.

2 IBM's WebSphere Micro Environment (WME)

WebSphere Micro Environment contains a production-ready Java Powered runtime environment; and a whole lot more. The environment has been tested and certified to meet the following J2ME specifications as specified by the Java Community Process[SM]:

For **small devices** (cell phones, Stinger based phones, small PDAs), WebSphere Micro Environment ships a platform that meets the following specifications, designed for very small memory constrained devices: Connected,
Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP).

For **larger devices**, such as larger PDAs, PDA Phones and hand held computers, WebSphere Micro Environment supports the Connected Device Configuration (CDC) and Foundation Profile. This can also be tailored down, to meet the specifications of device manufacturers.

2.1 Standards Compliance

WebSphere Micro Environment was the first virtual machine to support the new J2ME specifications, simultaneously, across 15 platforms. Unlike other virtual machine offerings, it provides a platform that is certified to the latest J2ME specifications, and is free and clear from complex, third party licensing. This allows device manufacturers to start today, to build devices that will host the next generation of Java applications.

IBM is committed to J2ME technologies and is listed as a J2ME licensee at the following Sun website: http://java.sun.com/j2me/licensees/ibm.html. WebSphere Micro Environment passes the Technology Compatibility Kit (TCK) testing as outlined in the Java Community Process, to earn the Java Powered logo. Java Powered is the designation for production level virtual machines that comply with the specification.
The advantage for customers is they have a production level product that has been pre-tested to the specifications, so that applications can be deployed to new devices quickly. This speeds the time to market for new devices, and allows each device to take advantage of multiple applications written to the specifications.

IBM is also ready to indemnify, in writing to its customers, its claims on compliance and to competitors' claims that third party licensing is required.

2.2 Memory and Speed

WebSphere Micro Environment was designed with the constraints of limited memory availability and processing power in mind. By building the virtual machine to perform under the constraints of cellular telephones, the underlying architecture easily scales to deliver performance and footprint capabilities needed for the next generation of PDA and smart phone devices.

WebSphere Micro Environment contains execution options that can help optimize the memory footprint for specific devices. It automatically allocates and de-allocates memory for full memory compaction. Tunable, interruptible garbage collection (GC) technology allows time-critical or deterministic events to halt GC. Precise GC ensures maximum memory compaction. Generational point and sweep garbage collection is also supported. Files can even be stored in flash memory and executed "in place" in ROM, minimizing application startup times and RAM requirements.

WebSphere Micro Environment also has software accelerators, tuned to each hardware platform, that take advantage of native operating system (OS) capabilities (threading, memory, floating point, graphics, etc). This ensures that WebSphere Micro Environment delivers performance while keeping memory requirements at a minimum. A unique, Ahead-of-time (AOT) compile technology can be used in end-user applications to take advantage of the same optimizations. WebSphere Micro Environment can be used to interpret bytecodes and also provides a Just-in-time (JIT) compile bytecodes to speed up generally available applications. The JIT and AOT compilers are also certified to the J2ME specifications listed above.

2.2.1 VM Size

IBM works with each of its customers to optimize both VM performance in relation to the amount of memory on the device. WebSphere Micro Environment can be tuned for exacting requirements for memory usage.

2.2.2 Performance Benchmarks

Independent results[1] confirm WebSphere Micro Environment has been proven faster than the competition (Jeode, KVM) using standard tests, Spec98JVM, Embedded Caffeine Mark, 10K iterations In all optimization modes, JIT versus Dynamic Adaptive Compile. A complete comparison of vm performance for the J9 virtual machine can be found at the North Carolina State University website at http://www.csc.ncsu.edu/embedded-java/

2.2.3 Virtual Machine Size versus Speed

With advancements in vm performance, there also needs to be consideration for vm size. Since optimization technologies differ from company to company, a small overview is needed.

[1] Independent Results http://www.csc.ncsu.edu/embedded-java/

Bytecodes are either shipped on the device, or are downloaded over-the-air to the device. The virtual machine is invoked, and the bytecode is interpreted on the device. This is the standard method in which java programs execute on small devices.

The next step in optimization is the just-in-time (JIT) compiler. A JIT compiler needs to be present on the device (adding to the footprint). As a bytecode is invoked on the device, it is compiled into a native executable (into machine code or native code – C/C++). This initial compilation step takes time degrading performance of the first invocation of a program. Since the code has been compiled into native form, subsequent invocations of the bytecode are speeded up.

Several companies have implemented various forms of dynamic adaptive or JIT compilation (IBM included), but the above premise must be followed. The key questions to ask vendors are – where does the compiler occur, and what is the overhead on the device for the compiler?

To take advantage of further optimizations that individual devices can deliver to Java applications, IBM has also implemented a native executable known as a JXE file (J9 Executable). Similar to an exe for a particular device, JXEs can take advantage of the devices H/W and RTOS to maximize performance. JXEs are created at compile time, by the developer, and target an specific H/W and RTOS platform (for example – PocketPC 2002 running on a StrongArm SA-1110). By pre linking JXEs to a particular platform, the virtual machine can take advantage of native threads on the device, use floating point accelerators offered by the RTOS or H/W.

WebSphere Micro Environment also has an Ahead-of-Time compiler that statically pre-links a bytecode, along with all of its dependencies, to achieve an even faster startup time, as well as avoiding the overhead of shipping a JIT compiler on the device. Since the compilation step is performed at development time, the overhead of compilation on the device (footprint and initial compile on the device) is avoided. Customers and developers can also take advantage of this technology on the platforms where it is supported. Although AOT compiled applications can only target a specific RTOS / HW combination, the

IBM uses this technology within WebSphere Micro Environment, on specific platforms to achieve VM performance. AOT code can be used in conjunction with both interpreted code, and code that has used the Just-in-Time (JIT) compiler.

The figure 2.2.3.1 illustrates the overall space versus speed tradeoff for interpretation, JIT, JXE and AOT compilation.

2.2.3.1 Figure: Interpreted vs JIT vs JXE vs AOT

		Mem	Speed (1, 2-N)	When to use
Interpret	Loads classes dynamically into VM			
JIT	Compiles bytecode upon loading			- Repitive code
JXE	Pre-links (dynamic) for Target, Optimizes			- Microanalyzer - Startup - Error Recovery
AOT	Pre-links (static) and optimizes app			- M/A identifies 5% - Startup code - Error recovery

2.2.3.2 Figure : Space versus Speed

	Startup (msec)	8 sec video	% CPU	Memory (mb)
Interpret	993	21	100	13.1
JIT	3315	8	74	22.7
JXE	420	14	100	13.3
AOT	354	8	36	12.6

Running 8 second MPEG 4 video (Emblaze) - x86 WindowsShows AOT improvement of startup time on x86

The above example, using an eight-second streaming video (figure 2.2.3.2), shows the tradeoff between space and speed. With dynamic or JIT compilation, the memory on the device is increased, since the compiler has to be resident on the device. Also note the lag in startup time for JIT compilation, since the bytecode has to be compiled on its first invocation. Using AOT and without the overhead of an on device compile, a combination of low startup time, fast execution, and low memory footprint can be achieved.

In version 5.0, IBM will implement the AOT functions on the ARM family of processors. This will bring new levels of optimization to ARM devices, and allow device manufacturers to take advantage of advanced optimization, without the burden of extra memory.

The following figure 2.7.2, illustrates the impact of JIT and AOT on the Embedded Caffeine Mark / MHz benchmark on the x86 and PowerPC platforms. The version 5.0 implementation of AOT on ARM will raise the ECM/MHz for the ARM platforms as well (benchmarks will be run upon delivery to the market).

2.2.3.3 Figure: Space Versus Speed – ECM/MHz – Impact of JIT / AOT

2.3 Native Interfacing

WebSphere Micro Environment supports the Java Native Interface (JNI), which allows you to directly access native (non-java) application interfaces, device drivers and OS functions. This provides maximum flexibility for original device manufacturers to select the peripheral devices (Point-of-sale, bar code scanners, USB devices, etc.) for industry specific applications, or just to meet the ever-increasing demands for differentiating functions.

Native threads (i.e., the thread capabilities of the underlying RTOS) are used when available from the underlying RTOS in order to utilize its highly tested and reliable scheduler. Native thread support ensures that the right code executes at the right time and in a well-behaved manner.

2.4 Connectivity

The major differentiating factors for selecting WebSphere Micro Environment is industry support and connectivity. The network of third party vendors reselling and integrating their products and services with WebSphere Micro Environment, all using the same open standards, ensures that manufacturers have expanded choice of features and functions to offer on their devices.

By pre-testing middleware and server connectivity, using open standards, WebSphere Micro Environment delivers the convenience of mobile devices, with the power of e-business. More than a stand-alone device for Personal Information Management and entertainment, WebSphere Micro Environment provides a platform where high value data services become a reality through relational database access data store and forward, transactions and application synchronization extensions. Connectivity to existing applications running e-business applications forms a complete end-to-end solution.

Proven over the course of five releases, WebSphere Micro Environment is already used in production to coordinate construction schedules, drill for oil in the North Sea, and deliver instantaneous messaging to mobile employees.

2.5 Management of Applications and Over-the-air provisioning

In Version 5.0, IBM will introduce elements from MIDP NG that allows for the provisioning of MIDlets to devices. This solution will compliment the work that IBM and its partners are providing in OAP provisioning servers.

WebSphere Micro Environment will also ship a beta implementation of a SyncML/DM client written to the CLDC configuration. This will enable the OTA synchronization of applications from the service provider to the handset. SyncML is the open standard that drives data mobility by establishing a common language for communications between devices, applications and

networks. SyncML Device Management (SyncML/DM) enables OTA administration of devices and applications, simplifying configuration, updates and support. This function will be premiered as a beta implementation in Version 5.0 of WebSphere Micro Environment.

2.5.1 Open Services Gateway initiative (OSGi)

For larger devices (utilizing Foundation / Personal J2ME profiles), IBM supports the Open Services Gateway initiative (OSGi). OSGi is an open standards group, committed to the over the air provisioning and maintenance of software. The OSGi framework works on bundles of software functions (classes, files,…) that can be pushed to, or pulled from, the device.

Bundles also contain a manifest list of all prerequisite and corequisite bundles that are needed for the application to function properly on the device. The manifest lists can be resolved (checked) before they are downloaded, to determine if the application can be started. Only resolved applications can be started. In some case resolution will include the provisioning of prerequisite or corequisite bundles to the device.

Bundles that are not active can also be removed from the device, or replaced on the device with new functions. In this way bundles can be provisioned to devices, updated on devices and removed from devices. Complete application life cycle management – all transparent to the service provider's customers.

IBM was the pioneer in the Component Distribution System (CDS), and extension of OSGi for device provisioning and maintenance. Elements of the existing CDS system supported by WebSphere Micro Environment will be introduced into the future versions of the OSGi specification. Customers investing in CDS will have a jump on the market.

2.6 Platforms Supported

WebSphere Micro Environment delivers right platform for building, deploying, testing, and maintaining e-business applications on small devices. Supporting Connected, Limited Device Configuration (CLDC™), Mobile Information Device Profile (MIDP™), Connected Device Configuration (CDC™), and the Foundation Profile™, WebSphere Micro Environment provides production-level environments for J2ME configurations and profiles that have been tested against J2ME specifications and certified "Java Powered™". These J2ME components have been simultaneously released—certified Java Powered—on an unprecedented number of target platforms.

A complete list of platforms supported can be found at www.ibm.com/software/pervasive

3 WebSphere Studio Device Developer - Integrated Development Environment (IDE)

To assist in creating the ultimate e-business device, IBM has a new member of the WebSphere Studio family of Application Development products; WebSphere Studio Device Developer provides functions that can allow Java developers to create, test and deploy e-business applications on small devices.

Just as WebSphere Studio Application Developer is popular with developers building J2EE™ applications for deployment to WebSphere Application Server; WebSphere Studio Device Developer can help create J2ME ™ applications that target WebSphere Micro Environment. Using the WebSphere programming model, it allows developers to create applications that enable cellular phones, PDAs and hand held computers to become part of an end-to-end e-business solution.

In creating WebSphere Studio Device Developer, IBM took the productivity features that made its VisualAge Micro Edition an award winning IDE and integrated them as plug-in extensions to WebSphere Studio Workbench. WebSphere Studio Device Developer features:

- SmartLinking technology that can optimize applications so that the memory of small devices is used efficiently.
- Just-in-time (JIT) and ahead-of-time (AOT) compilation technology that can allow developers to improve the execution time for portions of their applications that require special attention.
- Selectively can allow applications to be deployed to flash memory and executed in place (XIP), providing “instant on” capability for the end user.
- “On target” remote debug sets WebSphere Studio Device Developer apart from the competition by allowing the developer to deploy an application to a device and remotely debug the application as it runs, ‘live’ on the device. Changes can be made and tested immediately on the target device eliminating the time wasting cycle of editing, compiling, downloading and testing. This assists the developer in testing the application in the actual environment that it will be deployed into; critical when you are deploying applications to millions of devices.
- “On Target” MicroAnalyzer monitors the application as it runs on the device, and builds a log file of the methods called, the memory used, and the thread switches back on the developers workstation. This can allow developers to discover bottlenecks in application performance so that the application can be quickly tuned to deliver enhanced end user experience.

As applications deployed to millions of devices require updating in the field, the creation of bundles for deployment to an Open Systems Gateway initiative (OSGi) server and update on the device using either push or pull technology is needed. WebSphere Studio Device Developer simplifies the support of the Open Services Gateway initiative (OSGi) to deliver bundles of software to devices, and have the devices maintained through push or pull technology. WebSphere Studio Device Developer provides tools to help simplify the task of creating OSGi bundles.

Finally to assist in the creation of the ultimate e-business device, samples and examples combining the superior messaging of MQe and the advanced data management capabilities of DB2e are provided for the developer.

The WebSphere Studio Device Developer tools have been implemented using the Eclipse "Plug in" architecture (www.eclipse.org) and as such benefits from all the common services such as user interface, project view, help system, tool-to-tool integration, Java ™ IDE, edit, refactoring and the version control interface provided by WebSphere Studio Workbench. These common services present the development team a single, seamless, collaborative environment for the development of end-to-end e-business applications. WebSphere Studio Device Developer delivers productivity through integration.

WebSphere Studio Device Developer is available separately from WebSphere Micro Environment. It can be used to create applications that target any J2ME environment, but has features and functions that integrate closely with WebSphere Micro Environment. For a complete list of tool features and functions, please refer to appendix III.

For an evaluation copy of WebSphere Studio Device Developer, please visit www.ibm.com/software/pervasive

4 Conclusion

As applications are extended to mobile devices, new technologies will be required to provide productivity to the end-user while shielding them from the inherent complexities of integrating local and remote services. Local services such as barcode scanning, scalable vector graphics, full-motion video, and voice navigation will be required on the device. Remote services, residing within the service infrastructure, such as location (GPS), transactional and messaging services will need to be accessed wirelessly by the device.

The merging of local and remote services into a device that delivers the exact functions that are needed to deliver better customer service creates the ultimate e-business device. All of these technologies exist today. To successfully integrate and deploy them is a "simple matter of programming". To get there faster requires two things:

- The right platform
- The right tools

4.1 WebSphere Micro Environment - The Right Platform

In WebSphere™ Micro Environment, IBM® has extended the WebSphere software platform onto devices. Using the same programming model that has made WebSphere the industry-leading platform for e-business, IBM has created the foundation for the deployment of e-business applications to small devices. By using open standards to combine the portability of Java™ technology with the power of WebSphere, IBM helps bring the convenience of mobile devices to e-business. Together with the superior messaging of MQe, the advanced data management capabilities of DB2e™ and the scalability of WebSphere, IBM has created a complete platform for extending e-business onto millions of devices.

WebSphere Micro Environment provides the developer with a production level J2ME™ "Java powered"™ runtime environment, plus a whole lot more. It has been tested and certified to meet Connected Limited Device Configuration (CLDC™), Mobile Information Device Profile (MIDP™), Connected Device Configuration (CDC™), Foundation Profile standards.

The right platform must balance the device's memory constraints with the user's need for speed; critical for taking maximum advantage of battery powered devices where the user pays a per minute connection charge. At the core of WebSphere Micro Environment is the IBM J9™ virtual machine, which has been optimized for each supported device, helping to ensure application performance. This performance has been proven in independent testing.

Finally, what really sets WebSphere Micro Environment apart is that it provides the foundation for extending existing e-business applications to these convenience devices. By taking advantage of the middleware provided by WebSphere family of products, a complete platform for end-to-end device to services applications is provided. With proven performance, certified compatibility and the assurance of WebSphere, IBM WebSphere Micro Environment provides the right platform to mobilize an enterprise.

4.2 WebSphere Studio Device Developer - The Right Tools

IBM has also used the eclipse platform's plug-in architecture (www.eclipse.org) to integrate the award winning functions contained in its VisualAge™ Micro Edition into WebSphere Studio Device Developer. This new product, built on IBM's WebSphere Studio Workbench, provides developers with a complete build, deploy and test environment for the creation of Java applications that target WebSphere Micro Environment.

IBM brings the power of WebSphere software platform to millions of devices. WebSphere Studio Device Developer uses its history of award winning IDE features to help easily and productively develop and deploy java applications for WebSphere Micro Environment supported devices. Together they are the right platform and the right tools!

I.I Appendix I – WebSphere Micro Environment – Standards

- **Certified Java Powered** – http://java.sun.com/j2me/licensees.html
- J2ME Specifications Supported – Certified Java Powered to the following specifications
    - Handsets / Smartphones
        - Connected Limited Device Configuration (CLDC 1.0a) – JSR 030
        - Mobile Information Device Profile (MIDP 1.0a) – JSR 037
    - PDA / PDA Phones
        - Connected Device Configuration (CDC 1.0_0.1) – JSR 036
        - Foundation Profile – JSR 046
- **OS Support (Handsets/Smartphones)**: Linux, OSE, Windows CE, Windows Powered Smartphone 2002, BREW, REX, ITRON
- **OS Support (PDA, PDA Phones)**: PocketPC 2002, PocketPC2002 Phone Edition, PalmOS, Windows CE, Linux
- **OS Support (embedded)**: QNX, MontaVista Linux, Solaris soft kernel, AIX, Windows NT, Linux
- **Microprocessor Support**: ARM, StrongArm, Xscale, x86, PowerPC, MIPS, 68K, Hitachi
- **Class Library Support**: Full implementation of the classes contained in the following packages (including sub classes). Java Powered:
    - ***CLDC*** – java.io (system io), java.lang (programming language), java.util (collections, data/time, misc), java.microedition (corrections)
    - ***MIDP*** – java.microedition.rms (record storage), java.microedition.midlet (MIDLets), java.microedition.io (HTTP), java.microedition.lcdui (User Interface toolkit)
    - ***CDC*** - java.io (system io), java.lang (programming language), java.lang.ref (special reference classes), java.lang.reflect (reflection), java.math (math), java.net (network classes / tools), java.security (security), java.security.cert (certificates), java.text (text manipulation), java.util (collections, date/time support, misc.), java.util.jar (Jar file), java.util.zip (zip file), javax.microedition (corrections)
    - ***Foundation*** - java.io (system io), java.lang (programming language), java.lang.ref (special reference classes), java.lang.reflect (reflection), java.math (math), java.net (network classes / tools), java.security (security), java.security.acl (access control list), java.security.cert (certificates), java.security.interfaces (security interface), java.security.spec (key specifications), java.text (text manipulation), java.util (collections, date/time support, misc.), java.util.jar (Jar file), java.util.zip (zip file), javax.microedition (corrections)
    - ***Personal*** – Personal Profile provides an environment for applications that rely upon the AWT heavyweight components. It provides full AWT support (relative to JDK 1.1 AWT) and includes a number of feature improvements introduced by J2SE. This Profile serves as a suitable platform for web applets and as a J2ME migration path for PersonalJava applications.

      The Personal Profile is still in the JCP process, but advanced beta implementations are available, at the request of customers. IBM will certify the Personal Profile, as the TCK becomes available.

      Packages: java.applet, java.awt, java.awt.color, java.awt.datatransfer, java.awt.event, java.awt.event, java.awt.image, java, java.beans, java.io, java lang (programming

language), java.lang.ref (special reference classes), java.lang.reflect (reflection), java.math (math), java.net (network classes / tools),java.rmi (rmi), java.rmi.registry(registry), java.security (security), java.security.acl (access control list), java.security.cert (certificates), java.security.interfaces (security interface), java.security.spec (key specifications), java.text (text manipulation), java.util (collections, date/time support, misc.), java.util.jar (Jar file), java.util.zip (zip file), javax.microedition.io, javax.microedition.xlet, javax.microedition.xlet.ixc.

- **Device Management Support** – Open Services Gateway initiative – OSGi.org
- **Connectivity Support** – TCP/IP, RMI, 802.11b, JDBC, JMS, GPRS, USB, Serial
- **Tools Support** – WebSphere Studio Device Developer, an Integrated Development Environment (IDE) or any tool that supports the development of J2ME that and the following interfaces - Java Debug Interface (JDI), Java Virtual Machine Profiling Interface (JVMPI)
- **Partner Offerings** integrated using open standards
  - Secure Sockets - SSL – Wedgetail
  - Relational Database (JDBC) – IBM DB2e
  - Object Oriented Database (OODB) – Cloudscape
  - Application Development Tools – Eclipse Plug-in technology – eclipse.org
  - Messaging Middleware – JMS, IBM MQe, Applied Reasoning Mobile Classic Blend
  - Browsers enabled for the Java runtime environment – Opera, NetClue
  - Streaming Multimedia (MPEG4) – Emblaze, PacketVideo
  - Jini Networking (Jini) – PsiNaptic
  - Bluetooth – Rococco Software
  - Linux PIM applications – Trolltech QT/e

- **New in Version 5**
  - Native Browser integration (Pocket IE)
    - MIDP Over the Air (OTA) Extensions for carriers (MIDP Extensions)
  - Application Management – SyncML/DM (Beta)
  - Personal Profile (Beta implementation subject to TCK test suite availability).

I.II Appendix III – WebSphere Micro Environment – Features

WebSphere Micro Environment – V5.0		
VM Speed		
	Ahead-of-time compiler	Much faster (order(s) of magnitude), without the overhead of memory increase. Used primarily for startup code, error recovery code and for 5% developer code that is in need for performance boosting.
	JIT (Just-in-time) Compiler	Much faster (order(s) of magnitude), but requires JIT compiler on target - space / time tradeoff. Used for highly repetitive code.
	Adaptive JIT optimization	Frequently used code is compiled at higher level of optimization
	Can pre-locate/digest whole application	Reduces startup time & footprint significantly
	Fast JNI Implementation	Reduces time in calling C/C++ routines
	Fast Interpreter	12 years experience in building VMs
	Is the speed of the VM validated by third parties	http://www.csc.ncsu.edu/embedded-java/
	Optimized for ARM processor	AOT is new in version 5.0 for ARM
VM Size		
	JXE support	Compresses executables and resources into a single, easy-to-distribute, compact package.
	Supports XIP (execute in place)	Reduces RAM footprint significantly: fixed bytes stay in flash ROM, only variable part copied to RAM
	Class Libraries can be selected to span across device types. From extremely small to Internet Appliance Devices and gateways:	Size is critical for resource-constrained devices where cost per unit is key. Functionality needs to be scalable, and customizable.
	Tiny J9	Devices under 1MB - ie: Static ARM processor
	CLDC/MIDP - Static Arm processor	Library for minimal functions for limited devices (CLDC). Java.io, lang, util, x.microedition. MIDlet support.
	Full J9	Devices (2 Meg Plus) - SmartPhones, PDAs, Residential Service Gatways, Set Top Boxes, Commercial Gateways, Internet Appliance Devices - Advanced support for floating point, advanced GC (mark / sweep), finalization, Math lib enhancements.
	Supports PersonalJava 1.1 and 1.2 applications	These applications can be run, but are not optimized to J2ME. Applications written to the old specifications (JDK 1.1.8) will take considerable amount of refactoring to move to J2ME
	Customization of runtime to maintain compatibility, yet reduce the size of the footprint.	More room for the application. Less room on the target device for the runtime - available upon request.
VM Standards		
	Supports Latest J2ME Standards	Supports the latest JDK levels
	JDK 1.3 JRE levels	Newer, more stable base

	Certified CLDC (1.0a)	TCK Tested and certified compatible to the specifications as outlined in the Java Community Process(SM)
	Certified CLDC/MIDP (1.0a)	"
	Certified CDC (1.0_01)	"
	Certified CDC/Foundation	"
	J2ME CDC/Foundation/Personal Profile (NG)	Awaiting TCK for certification, beta level only
	Web Update for new specifications as they become available	Having a web update feature to distribute the latest tested profiles and configurations, as they become available (MIDP NG),…
	TCK Tested to earn the Java Powered(TM) logo.	Product, not reference implementation. Supported. Free and clear of all third party licensing
(debug)	Supports JDI (Java Debugging Interface)	Can use standard debugging tools
(debug)	JDI code on host or remote target	Code on host saves space / increases speed on target
(profiler)	Supports JVMPI (JVM Profiler Interface)	Can use standard tools (I.e. JProbe)
VM Special Features		
	Supports Multiple Memory Configurations (flash, ROM, discontiguous)	Flexibility to choose the device that is right for the application based on speed, cost, maintenance and upgradability
	Run Multiple VMs concurrently (per thread, per process)	Allows application isolation and flexible RTOS process control
	Execute from ROM	Useful when you want to offer additional aftermarket parts, services to a device that has shipped
	Allows multiple VMs with shared or separate stacks	Allows memory Isolation
	Single-source JCL and VM	Consistency across all target platforms for portability of application, eases migration and delivers uniform execution environments
	Structure with portability layer for the application and optimization layer for the platform	Portability layer for applications allows multiple platforms to be targeted Optimization layer takes advantage for each platform for speed,…
Memory Management		
Garbage Collection (GC)		
	Realtime extensions for deterministic execution	Threads can execute at higher priority than GC for critical tasks
	Precise or conservative?	Precise = better (Conservative = not all objects collected)
	Configurable / tunable?	Improves speed & space
	Incremental	Reduces minimum timeslice needed/GC does not lock VM (needed for realtime)
	Interruptable	Required to handle time-critical interrupts & tasks
	Generational	Scanning of generational objects saves lookup time and increases CPU efficiency.
OSGi		
Development Environment		
	Test Environment provided / available - for development use only	Needed for OSGi development

Flash support		
	Can write bundles directly to flash	Enables component update and activation without interupting the customer
	Can execute downloaded code in place (XIP)	Saves RAM - fixed bytes stay in flash ROM, only variable part copied to RAM
Natives		
	Supports download of native code, drivers, (non-Java) files	Standardizes downloads, allows config management of non-Java files
IDE support		
	Built in bundle upload to server	Greatly simplifies OSGi server management
	Debug Support for bundles	Debugging bundles avoids Printf debugging
	Assists in defining prerequisites	Simplifies complex bundle / package relationships, helps debug prereq issues
Other		
	Follows prereq chain	Ensures required prerequisites are also downloaded, even if not specified by client
GUI		
AWT		
	AWT / SWT standards-based UI for Java	Allows interoperability of apps - however, significantly larger and slower than bitmap-based systems: full windowing controls
Browser Integration		
	Browser Integration across multiple platforms	Has partnerships with Browser companies who support browsing across multiple platforms (NetClue, Opera)
	Browser Integration on popular devices (Pocket IE)	Reduces memory required when original brow ser on device is used - This will be shipped via web update.
Connectivity to Data Services		
Carrier		
	Provides a J2ME Java Powered runtime environment	Device becomes a platform for multiple applications written by different developers. Opens up device to applets, midlets,... written by the development community.
	MIDP Extensions - Can be modified for individual carrier extensions.	Each carrier has their own communications stack, security, HTTP and provisioning engines.
	SynchML/DM synchronization engine for	For synchronization of files, applications, PIM data and data services
	CLDC/MIDP	
	CDC/Foundation	
	Integrates with provisioning servers	
	BREW	
	JSR 124 Generic Provisioning Servers	Upon Request
	Extensions for individual carriers prov isioning engines	Upon Request
e-business		

	Integrated with device databases using Open Standards (JDBC)	DB2e
	Integrated with Ojbect Oriented databases	Cloudscape
	Integrated with messaging middleware	
	MQSeries / WebSphere MQ	Asynchronous Messaging
	JMS	Asynchronous Messaging

I.III Appendix III – WebSphere Studio Device Developer

	Feature	Benefit
WebSphere Studio Device Developer		
Editor		
	WebSphere Studio Integrated toolset	Shortens coding cycle time, errors
	Code-assist / auto-completion	Simplifies coding / reduces manual lookup time / ease of learning
	Compiles against selectable Class Libs (JCL's)	Allows switching from large to small libs, and gives immed. feedback
	OSGi Bundle creation support	Simplifies OSGi bundle definition & management
	Extensible / open plug-in architecture	Upgradeability, 3rd-party & customer customization
	Provides integration points to server side tooling	Plugs into WebSphere Studio Application Developer for an device-to-server build, deploy, test solution
	Programmable editor	
	Refactoring Tool	Important when migrating applications to take advantage of new specifications. Useful for migrating PersonalJava applications to J2ME
Debugger		
	Integrated into IDE	Simpler, faster, greater functionality
	On-target cross-platform debugger	Remote debug to multiple targets allows developers to test code as it runs on the target device. Debug client runs on client avoiding load and space utilization on the target
	Debugger does not alter executable (no hooks in application)	Testing actual application speeds the development process and ensures developer that app will run on target devices
	Hot code replacement (local)	Do not need to restart debugging session - saves lots of time
	Hot code replacement (remote target)	(cont'd) also don't need to re-download updated code
	Plug in for PalmOS Emulator	Assist in local testing of application on developers workstation
	Integrated PocketPC Emulator	Assists in testing of applications local of developer workstations
	Integrated BREW emulator	Assists in testing of "TRUE BREW" applications
MicroAnalyzer		
	Integrated into IDE	Simpler, faster, greater functionality
	Memory space monitoring	Needed for debugging memory use issues, leaks, GC issues, etc.
	Logic trace-like display shows thread switches	Critical for debugging thread issues, deadlock, timing issues, etc.
	shows JNI calls	Required for testing native code calls
Repository - Team Streaming Technology		

	Differences Browser	Allows developers to compare and quickly merge code changes
	Supports popular Version Control (VC) products	Integrates with CVS (open version control system). Also plugs into WebSphere Studio Application Developer for access to more Source Code Management Systems (Rational, TogetherSoft, Merant,…).
	Supports multiple distributed servers easily (WAN)	Projects can be distributed, and one developer can work on serveral projects for several teams across network (employees / suppliers/ ...)
	Replication between servers or desktop	Simplifies development when disconnected from network. Only file changes have to be transferred so slow links can be used.
	Versioning	Can always immediately recall earlier versions applications for support or modification.
	Releasing	Used to share finnished code with teammates
Build / Launch / Test Integration		
	Integrated with ANT build scripts for smoother workflow between edit, build and test phase	Rapid build, speeds testing of applications. Tight integration with the version control systems, allows mutliple new ideas to be tested, and versionned together.
	Integrates with PalmOS Emulator (POSE)	Assists developers in quickly building applications for deployement to PalmOS devices
	Integrated PocketPC Emulator	Assists developers in quickly building applications for deployement to PocketPC devices
	Architected Emulator Interface allowing other skins to be quickly added	Allows new emulator skins to be quickly added to the development environment
	Integrated BREW SDK and emulator	Assists developers in quickly building TRUE BREW applications
	MIDlet skins for creating MIDP applications	Assists developres in creating MIDlets for deployment to J2ME Java Powered Runtimes
Open		
	Eclipse.org "Plug in' architecture	Uses eclipse.org architecture to allow other tools vendors easilty integrate.
	Integrates with C/C++ support	Stores C/C++ code in repository for VC of entire application
	BREW Development "Plug in'	Assists developers in creating TRUE BREW(TM) applications