Data Management Systems for the PostPC Era

Understanding the Business Benefits

Cloudscape

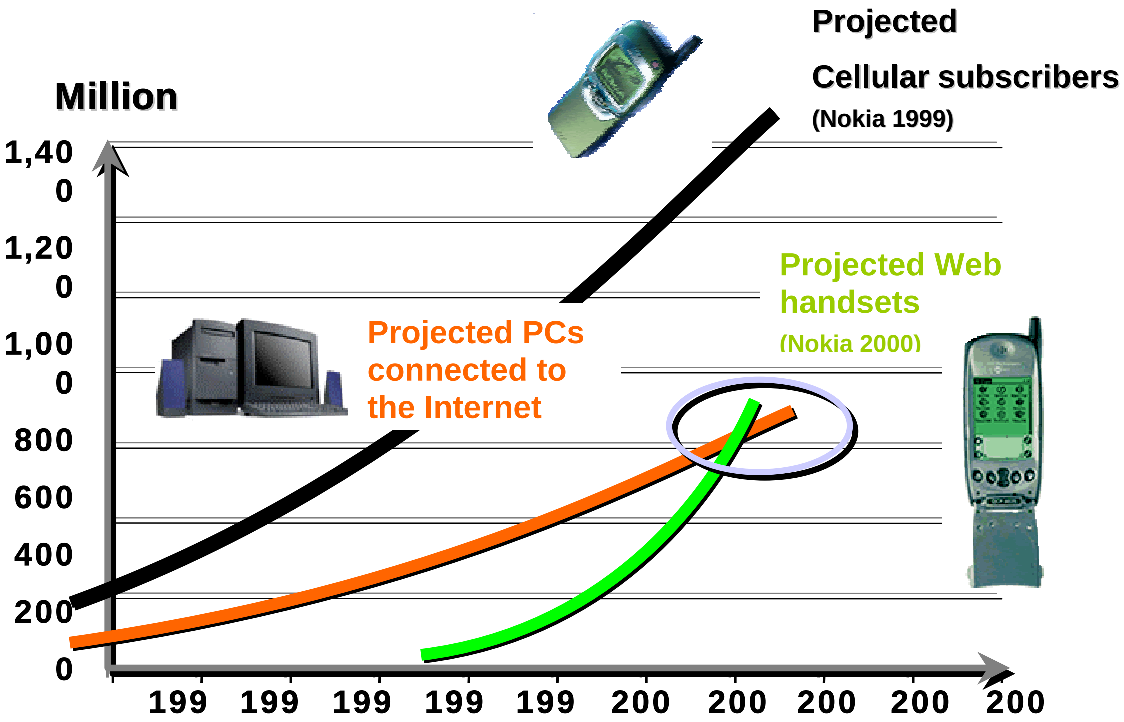
Introduction

As we start the 21st century, we are about to witness one of the most explosive unions of technologies in history.

The first technology is the Internet, which has already proven its ability to cause revolutions. It has, for example, already changed the way we think about business. Instead of thinking about how to integrate value chains within companies, we now think about how to integrate value chains *between* companies. We now look at how to provide new shopping experiences for consumers. And we've now realized the goal of sharing data between organizations without having to re-key it, thanks to the World Wide Web.

The second technology is mobile telecommunications, where another revolution is now occurring. Even as huge land-based infrastructures are being constructed to deliver broadband information, Third Generation (3G) mobile technology is poised to dramatically alter the capabilities of portable telecommunications. By 2003, estimates suggest that there will be more than one billion mobile phones in the world (Source: *Gartner Group).* And by 2003, this market will be growing by 50% per year, increasing to 100% per year by 2006.[1]

This growth represents change on a new scale. From a standing start in 2000, Internet-enabled mobile phones and Internet appliances of all shapes and sizes will be more commonplace than television sets—within just 36 months.



More handsets than PCs connected to the Internet by the end of 2002.

Figure 1: Handheld devices on the Internet outnumber PCs (Source: Nokia 2000)[1]

Until recently, the desktop PC has provided the primary access to the Internet, and the laptop PC has provided mobile access. However, as shown in Figure 1, handheld devices will soon provide the primary means of connecting to the Internet. We are now entering what we call the *PostPC* era, an era in which people access the Internet from a variety of devices—wired and wireless, mobile and stationary. From your mobile phone or other Internet appliance, you can already check the weather and the latest news, download flight schedules, do your banking, check on the stock market, order and pay for goods, and access many more services.

The PostPC era will impact how business is done by making information accessible to anyone, anywhere, at any time. This shift in how data is accessed and used requires a different approach to managing data. This paper discusses the role of database technology in the PostPC era, reviews product requirements, and takes a detailed look at Cloudscape™, a Java database product designed explicitly for the needs of this next generation of business applications.

Data management in the PostPC era

Database systems must evolve to suit the computing paradigm of the day. For the past 25 years, each new generation of computing—from mainframe to client/server to three-tier—has required new database technology to solve new problems. The change from mainframe to client/server computing, for example, created the need to manage data on two or more machine architectures, including Windows/Intel PCs and a host of multi-user computers.

The PostPC era will also bring about new problems for developers of business applications to solve. In contrast to applications of preceding eras, PostPC applications will need to:

- Work when only occasionally connected
- Support multiple heterogeneous devices
- Provide a seamless data experience for users
- Allow remote installation and maintenance
- Ensure data consistency and security

Being occasionally connected

In the PostPC era, applications designed for mobile use must work offline as well as online. Being frequently disconnected is the rule—not the exception—for several reasons:

- **Lack of coverage:** Despite several years of building cellular infrastructure, wireless services still suffer from unreliable access. Wireless network service can be spotty inside many buildings. In addition, wireless networks do not serve many geographic locations very well, so remote areas often lack coverage. Particularly for companies deploying a nationwide application or service, this lack of coverage means that some users will almost always be out of coverage for some part of the day.
- **Cost of service:** The cost of wireless Internet services often includes charges based on the amount of data transmitted. To reduce network charges, application developers may

create a local cache of commonly-used data so that only new data is actually transferred between the client device and a centralized application.

- **Performance:** Even with improved bandwidth, wireless network services are sometimes very slow, forcing a customer or an employee to wait long periods of time to complete a transaction. The cost of such delays is obvious—they reduce both customer satisfaction and employee productivity.

For these reasons, most business applications in the PostPC era must function even when not connected, so business applications based on thin-client-only access are not an option. Instead, the best option is to develop applications that function locally but can synchronize with a centralized data source when the timing is right. Data Management Systems in the PostPC era provide this synchronization.

Device heterogeneity

Tomorrow's new intelligent devices will revolutionize the way we live and conduct business by giving us instant access to information anytime, anywhere, regardless of where it's stored. As a result, PostPC applications may need to work with vast array of devices, as shown in Figure 2.

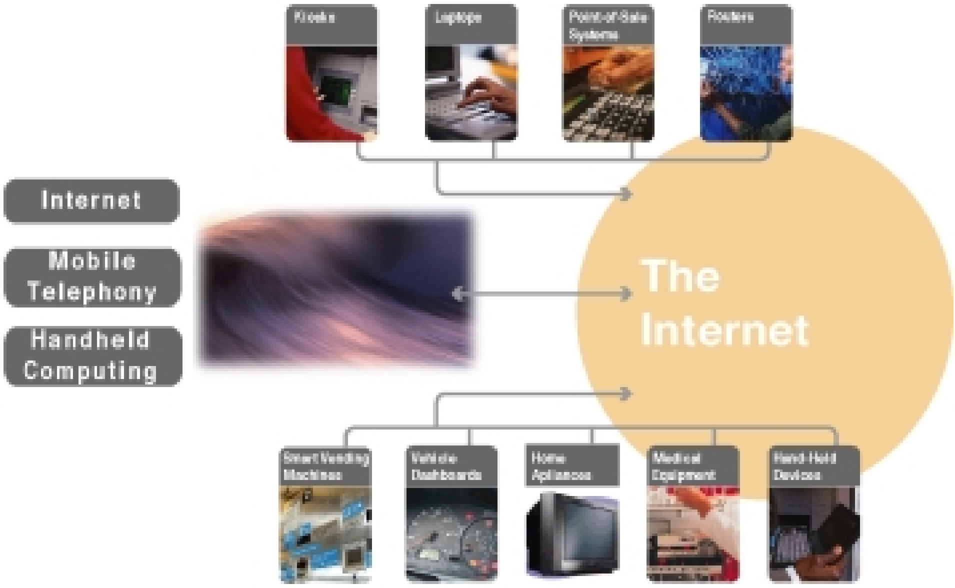


Figure 2: Internet access in the PostPC Era

With technology changing so fast and acceptance of technology accelerating, business line managers and CIOs must ensure that their applications are quickly adapted for lucrative new markets and popular new platforms. Consequently, when developers create applications intended to work across the vast spectrum of Internet devices, they need allow for changes in technology in order to "future proof" their applications. One strategy for future proofing applications adopted by many developers is to use the Java™ programming language and 100% Pure Java™ environments to deploy new applications. Later in this paper, we will investigate how Data Management Systems can be used to create these kinds of applications.

Seamless data

Today's users demand a seamless data experience—they want to be able to access the data they need, no matter which device they may be using at the moment. So a given application may need not just to support access through a variety of devices but to support access *by a single user* through a variety of devices.

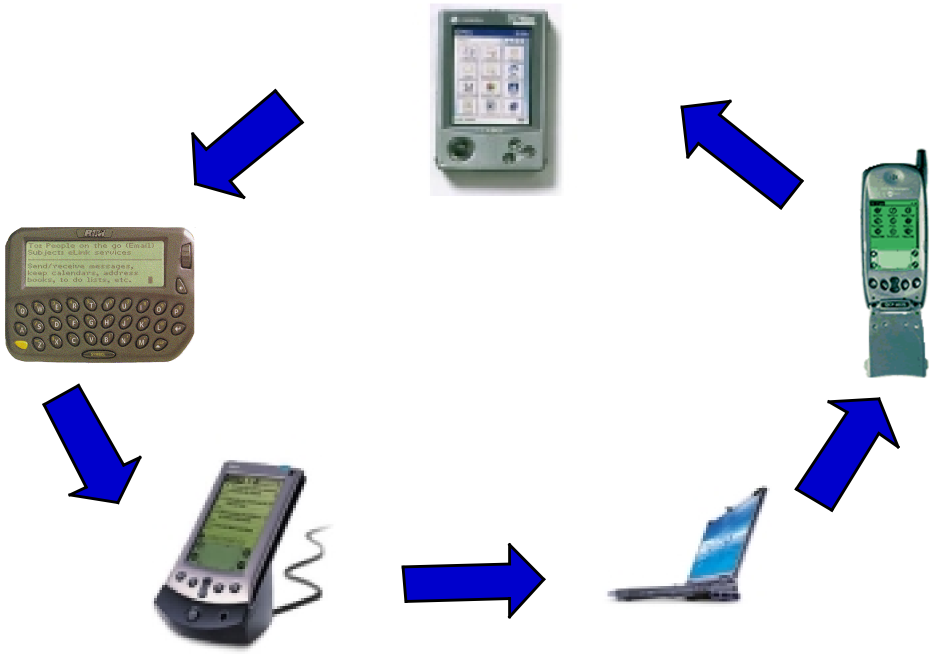


Figure 4: Users demand seamless data access regardless of their choice of device

Data synchronization is key to providing this seamless data experience. The file-based synchronization available today is insufficient because it is an all or nothing proposition, useful for exchanging documents but poorly suited to managing business data. In the words of one industry analyst, "It's tedious, arcane, inefficient, and counterintuitive."[2]

In today's world of business data, users typically need to see information that is relevant to them at a given point in time. So for mobile computing to advance, users need a way to publish and subscribe to the exact part of the database required for their jobs. By using powerful query languages such as SQL, developers can express how to subscribe to certain data sets and how to publish data from users' devices to centralized databases. By basing data movement and synchronization on database technology rather than on document management technology, developers can create a single centralized database that serves several thousand users.

Remote installation and maintenance

Devices such as cell phones, PDAs, and smart devices that communicate through the Internet already number in the hundreds of thousands, and, as shown in Figure 1, Web handsets alone will soon outnumber traditional personal computers. In addition, these Internet-enabled devices will have only a minimal set of administrative tools for managing data and applications, and the typical users of these devices will not be computer literate and will have no interest in the day-to-day management of applications and data.

Consequently, the cost of managing these devices must be miniscule in order for service providers to turn a profit. Similarly, organizations that deploy large mobile or remote applications must find economical ways to upgrade their applications, or they risk seeing those applications fall into disuse as their business evolves.

To meet these challenges, PostPC applications and data must be managed automatically from a centralized source, providing a zero administration remote environment.

Need for data integrity and security

Business applications have always needed to ensure data integrity. But in the PostPC era, applications will face special problems created by devices that are either mobile or stationary but remote. For example, basic business application design requires that transactions such as purchases of goods, accounting transactions, and shifts in inventory must leave databases in a consistent state. However, since mobile devices may move into and out of coverage several times a day, PostPC business application design requires transaction-based data integrity and consistency checking to capture vital business information.

In addition, devices located in remote areas and portable devices such as handheld computers may require extra security measures not needed for traditional applications. Since the data being transferred is often proprietary, developers need to take extra steps to ensure the security of this data even if the device itself falls into the wrong hands.

Cloudscape: A Java Data Management System

In the PostPC era, successful applications will take a new approach to managing data. The next generation of device-independent, distributed applications will be based on three critical technologies:

- The Java language
- SQL database management systems written in Java
- Application synchronization written in Java

Together these technologies comprise a Java Data Management System (JDMS).

Java language

Java provides three critical features that are required for PostPC applications:

- A platform for deploying applications that is largely independent of hardware and operating system

- A common object model
- A language suited to the representation of business logic

Machine-independent platform

Within an organization, developers can usually count on standardized platforms, but when creating applications to be deployed to partners, they often face a bewildering array of target machines, ranging from handheld PDAs, to laptops, through all flavors of Unix, to the AS/400 and even mainframes. Java provides a platform abstraction that insulates Java applications from this underlying complexity. This abstraction makes it possible to create a single, platform-independent version of an application that will run on any machine and operating system a partner or customer has chosen. This reduced complexity can result in significant cost savings.

Common object model

In Java, the common object model makes it possible for any Java application to work with any other. In C++, by contrast, programmers must agree on the compiler and the memory model before they can have confidence that two programs can cooperate. The Java common object model makes it possible to invisibly embed one application inside another, which in turn makes it easy to create database-enabled applications for situations where installing a separate database would be impractical.

Business logic language

Java is emerging as the business logic language of choice for new applications. Maturing standards such as Enterprise JavaBeans™ (EJB) can encapsulate business logic in a way that promotes developer productivity and code reuse. With Java providing a two to five-fold improvement in developer productivity over C or C++, IS departments are turning to Java for business logic, retaining C only for infrastructure or “systems” programming.

Java SQL DBMS

Some early attempts to deploy applications relied on the file system for storage. While a file system may provide an adequate storage mechanism for the simplest of applications, it lacks transaction support, query ability, and standardization across platforms, so it is likely to prove to an obstacle to future application enhancement. Although it’s possible to build sophisticated data management on top of the file system, it’s expensive to do so, and few IT organizations have the skills required to build a robust and portable data manager.

However, it’s possible to solve all these problems simply by plugging in an embeddable Java database management system (DBMS). Since most corporate applications rely on a SQL DBMS, the Java DBMS needs to be *relational*. Cloudscape is the only 100% Pure Java SQL RDBMS available from a major database vendor.

Application synchronization

The synchronization requirements of a deployed application go well beyond the capabilities of data replication as it is generally understood today. Replication-based solutions rely on moving data between copies of a database, with remote changes being merged into central systems, or peer systems merging changes with one another. In contrast, synchronized applications are designed for a hub-and-spoke architecture, as shown in Figure 5. This architecture establishes a clear single owner not just of the data but also of the business rules that may be applied to the owner’s copy of the data.

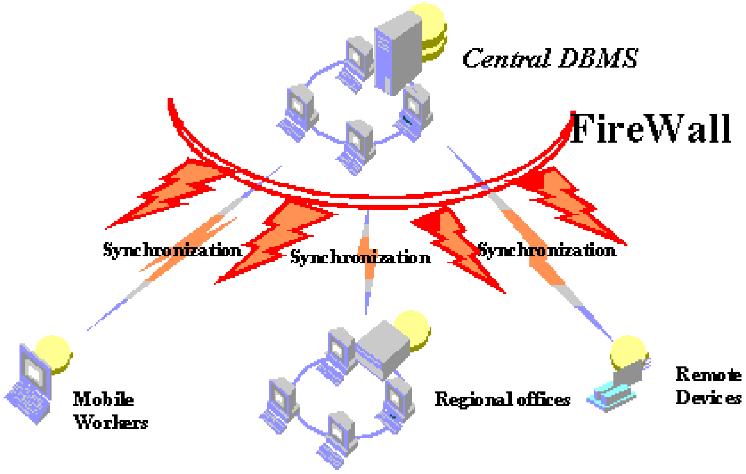


Figure 5: Hub-and-spoke architecture

In a synchronized application, the central application and database retain absolute control of the data and logic. In such an application, it is rarely appropriate to merge remote changes directly into the central database. Instead, remote changes must first be validated by the central application. This asymmetry of control demands an asymmetrical architecture, in which the central site both distributes data and logic to remote sites and also collects business events from those remote sites to be replayed through central applications. For example, a supplier may relay the information that it is about to fulfill an order. This information cannot be merged directly into the central database but must pass through the order entry system for validation, invoicing, inventory control, and more. On the other hand, new product information published at the central point needs no validation by the remote site: it is correct by definition.

A deployed application is made up of logic as well as data. A full application synchronization architecture must take this into account and manage the synchronization of application logic and schema changes, as well as the data itself.

Understanding Cloudscape

The Informix® Cloudscape product line is the industry leader in Java data management. It provides two critical technologies designed for building PostPC applications:

- The Cloudscape Java DBMS
- Cloudscape Application Synchronization, using Cloudsync™

Cloudscape DBMS

Cloudscape is an object-relational DBMS written in 100% Pure Java. Because Cloudscape is written entirely in Java, it can be invisibly embedded within a Java application, and it can also be embedded in a Web browser using the browser’s plug-in interface. A single shipping version of Cloudscape runs on most Java-enabled platforms, from today’s more powerful PDAs to mainframe computers.

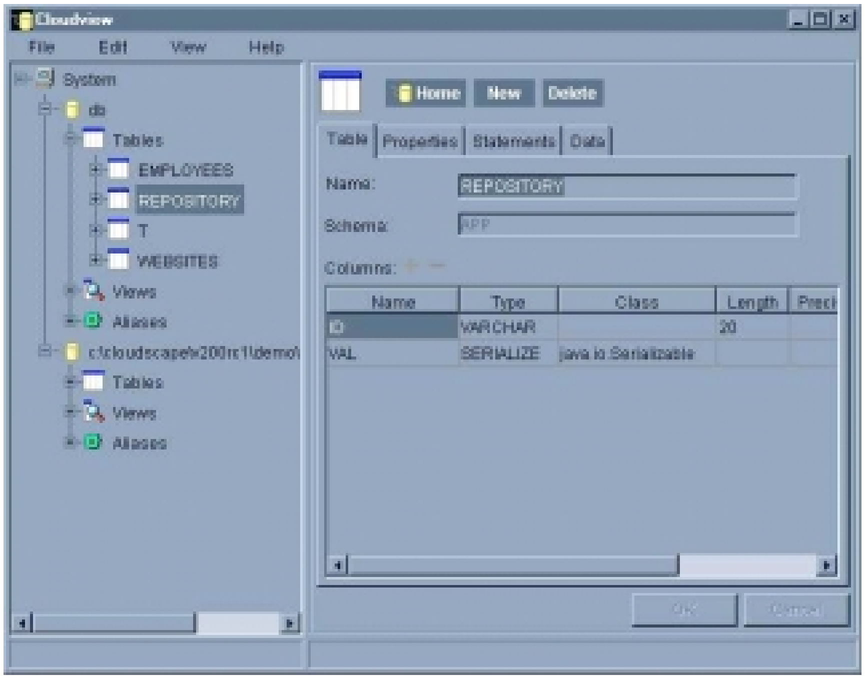
Cloudscape implements the SQL-92E standard with Java extensions. In Cloudscape, a developer can create a column in a table in which to store a Java class. The class will be stored as an object whose public methods are accessible from SQL, and those methods will behave as Java stored procedures running on the same Java Virtual Machine (JVM) as the DBMS itself. Cloudscape ships with a Pure Java (Type 4) JDBC driver, and a JDBC-ODBC bridge for ODBC applications is also available. Cloudscape databases can secure data by encryption on disk, encryption of the synchronization data transfers, and LDAP-based authorization.

Designed for deployed applications, Cloudscape does not require local management and does not require the user to have any database administration knowledge. It has been embedded in such diverse applications as product catalogs, application servers, email servers, supply-chain management software, and more.

Cloudsync

Using Cloudsync, Cloudscape Application Synchronization ensures the propagation of data from the central site to remote applications, moves business event calls from the remote applications back to the central site for replay through corporate applications, and manages the transparent deployment of application logic and schema upgrades published at the central site.

Through Cloudsync and Cloudview™ (the Cloudscape system management tool), the central site can handle security, authorization, and general management of the deployed applications and can also create publications to which remote applications can subscribe.



Name: REPOSITORY

Schema: APP

Columns:

Name	Type	Class	Length	Preci
ID	VARCHAR		20	
VAL	SERIALIZE	java.io.Serializable		

Figure 6: Cloudview system Management Tool

Cloudsync acts as a staging point for deployed applications. It stores data destined for remote applications and responds to refresh requests from remote applications, managing the download of data and upload of events. In contrast to the simple data synchronization offered by many DBMSs, Cloudsync is unique in its ability to synchronize the whole application environment—the data, the schema, and the application code itself!

Understanding the business benefits

Table 1 provides an overview of the benefits of Java Data Management Systems when compared to traditional technology.

Feature	Function	Benefit
100% Pure Java	Instead of merely allowing a database to be accessed by Java, a Java-based DBMS and synchronization system enables the whole application to move from platform to platform.	Provides 100% portability, which means: • Quick movement to new platforms • Reduced development expense • Less risk of choosing the wrong platform • Less dependence on a single platform vendor, more competitive pricing
Data synchronization	Enables applications deployed locally to share data with central databases.	Enables deployment of applications designed to be occasionally connected, which: • Increases application reliability and availability for mobile workers and distributed applications • Reduces network cost • Increases performance
Application code synchronization	Enables the database to manage all aspects of an application—unlike traditional systems, which can manage replication of data but can’t help when the application or the database schema changes. Furthermore, the synchronization of the code with the latest data keeps the application consistent even though the upgrade may occur over an extended period of time.	Enables remote maintenance, which: • Reduces the cost of maintaining the system • Allows quick upgrades for applications, creating higher customer satisfaction • Creates faster deployment of new services and applications
Publish and subscribe synchronization	Enables a user to subscribe to a particular service (defined in the system) that identifies a subset of the database relevant to that user or application. A given subscription could apply to multiple types of devices.	Creates a seamless data experience so that users: • Will receive the same data regardless of which devices they use • Can switch devices without losing context

Encryption and authorization	Enables data in Cloudscape databases to be secured by encryption on disk, encryption of the synchronization data transfers, and LDAP-based authorization.	Creates a secure data environment so that companies can: • Protect proprietary information even if devices are lost or stolen • Protect proprietary transaction data during synchronization • Use enterprise-wide authorization systems to validate a user's security privileges
Work-unit-based conflict resolution	Enables developers to specify how data collected remotely will be validated in the central database system during synchronization.	Ensures higher data integrity, which: • Lowers the cost of maintenance • Creates higher customer satisfaction

Table 1: Features and Benefits of Java Data Management Systems

Return on investment analysis

Ultimately, the return on investment for technology depends on the benefits of the application being deployed and the cost of the resources. These benefits and costs will vary greatly from application to application. However, certain variables should be taken into consideration in the justification of a Java Data Management System:

- Cost of network data transmission
- Impact of performance
- Impact of mobility
- Remote administration

Cost of network data transmission: For many mobile applications, the cost of the network service is the largest cost. When weighing the benefits of thin-client versus smart-client systems, application designers should consider the cost of transferring all the information required to perform the application versus the cost of storing commonly-used data remotely. Also, if users can synchronize their data at off-peak times, they can significantly reduce network costs.

Impact of performance: The major costs of poor performance fall into two categories:

- **Impact of poor performance on employee productivity:** To get a simple calculation of this impact, estimate the reduction in employee productivity, and multiply by the fully-loaded cost of each employee affected.
- **Impact of poor performance on customer satisfaction:** When poor performance affects distributed applications that directly support a customer or are required to perform a service on behalf of a customer (e.g., field service automation systems), then the financial impact should be measured in terms of a company's revenue and its reliance on retaining customers to make its revenue goals.

Impact of mobility: As with the impact of performance, the impact of mobility affects both employee productivity and customer satisfaction, so consider these questions:

- Does creating an application that enables mobile workers improve their productivity and ability to provide quality service?
- What is the impact of making this application unavailable for a period of time?

Remote Administration: Administration and maintenance of distributed applications is often a hidden cost for first-time users. One of the biggest impacts of Java Data Management Systems is the ability for a distributed application to be automatically and remotely maintained over time. Without this feature, many applications cannot be cost justified. When estimating the advantage of this feature on your mobile or distributed application, consider the following:

- What is the cost of deploying application updates manually (by sending a CD or memory upgrade to each remote site or portable device, for example)? What does it cost in productivity for your IT group, end users, and technical support group? Would your end users have the skills required to apply these updates?
- What is the cost of delaying a bug fix for critical problem?
- What is the cost of delaying the deployment of a new release of your application or service?
- What is the cost of being stuck on an antiquated platform for an extended period of time?

Remote administration scenario

The impact of these variables varies greatly from company to company and application to application. For instance, consider the following scenario, which compares manual application of updates via PC to synchronization based on remote administration:

Suppose that your mobile field service application is deployed to 1500 employees nationwide. You expect that your application will need minor bug fixes twice a year and a significant application upgrade once a year.

The cost of upgrading manually by distributing CDs to the field and then having users apply upgrades using PCs to synchronize can be estimated as follows:

> Assume creating software, burning CDs, and mailing CDs three times a year would cost approximately $15,000.
>
> Assume 20% of the field force would need telephone support when applying the upgrade. If the typical cost for a telephone service call is $100, then:
>
> > 1500 employees X 20% = 300 calls
> > 300 calls X $100/call = $30,000 per upgrade
> > 3 upgrades/year X $30,000/upgrade= $90,000 per year

Assume that employee time lost to perform upgrade (includes powering up PC and reading directions) = 30 minutes X 1500 employees. If the burdened cost of a field technician is $150 hour, then:

30 minutes @$150.00/hr = $75 cost of time lost per employee per upgrade
1500 employees X $75 = $112,500 cost of time lost per upgrade
3 upgrades/year X $112,500/upgrade = $337,500 total cost of time lost per year

The total yearly cost would then be:

$15,000	Cost of creating and distributing upgrade CDs
$90,000	Cost of support for users applying upgrades
$337,500	Cost of time lost applying upgrades
$442,500	Total cost of upgrades for one year

In this scenario, costs for manually maintaining the system would be $442,500 a year. Over the life an application lasting 5 years, the cost would grow to $2.2 Million.

By comparison, the costs of remote administration should be negligible since updates would be done as part of the synchronization process that workers performed regularly. These cost estimates are probably conservative but illustrate the impact of a remote application and data administration capability.

Summary

When building new eBusiness applications, developers need to evaluate database products carefully to ensure that the products can create applications that satisfy the additional requirements imposed by PostPC era. As this paper has shown, applications in the PostPC era must:

- Work when only occasionally connected
- Support multiple heterogeneous devices
- Provide a seamless data experience for users
- Allow remote installation and maintenance
- Ensure data consistency and security

The Cloudscape Java Data Management System has been designed with these requirements in mind. For organizations committed to the Java application promise of "Write Once, Run Anywhere™," the Cloudscape system is the JDMS product of choice.

References

[1] When Worlds Collide

Informix Corporation, 2000

www.informix.com

[2] Nokia in the 3rd generation

Nokia Networks 2000

FIN-00045 NOKIA GROUP, Finland

www.nokia.com

[3] Say Goodbye to the personal computer and hello to personal dataspace

Infoworld, Sept 8, 2000

www.infoworld.com