

## Addressing the Challenges of Updating Software and Firmware on Mobile Phones Over the Air

### The Growing Challenge of Mobile Handset Software Quality and Reliability

Over the last several years, mobile telephones have undergone a transformation from relatively simple voice-based handsets to more sophisticated multimedia communication devices used for a wide range of personal and business applications. Mobile phones now commonly include voice-enabled dialing, media players, rich messaging applications, web browsers, integrated cameras and other sophisticated features.

In order to provide this broad array of features, mobile handsets require increasingly complex software and firmware. The growing complexity of software, however, presents a significant challenge to mobile handset manufacturers and mobile telephone carriers: the ability to ensure quality and reliability. Given the volume of mobile handsets shipped — over 900 million units are expected to ship in 2006 — and the diverse conditions in which they are used, as software complexity increases, the likelihood of defects causing an operating malfunction increases.

To address this issue of software quality and reliability, mobile handset manufacturers have, up to now, relied primarily on a rigorous quality assurance (QA) process. Though necessary, this extended QA process has the significant drawback of delaying the release of new devices to the market, thereby exposing the manufacturer to a loss of market share.

Moreover, it is becoming increasingly difficult to ensure that the QA process is 100% reliable. Over the last several years, there have been multiple reports that handsets with seriously flawed software do get to the market. And there have been reports of devices running software that is vulnerable to viruses and other unauthorized access. The typical response from manufacturers and carriers is to require that users return the devices to a service facility for repair or replacement. This is an expensive and inconvenient process for the manufacturers, the carriers and the users. According to various industry sources, these recalls have been estimated to cost mobile handset manufacturers and carriers more than \$5 billion annually and affect more than 80 million phones.

As mobile devices and the software that operates them become more complex, what processes and technologies will be used to ensure that they operate securely and reliably?

### An Effective Solution: Firmware Over-the-Air Updating

To address the growing challenge of providing higher quality and more reliable software for mobile devices, eliminating expensive recalls, while at the same time meeting aggressive time-to-market demands, a more effective technology known as “firmware over-the-air” (FOTA) updating is gaining broad acceptance in the mobile telephone industry. FOTA updating enables mobile device manufacturers and carriers to remotely update the software and firmware that controls the function of the handset wirelessly. New software that patches flaws in the software originally installed on the handset is delivered over the air, eliminating the need for the user to bring the handset to a service facility.

In addition to delivering patches to fix faulty software and firmware, the FOTA updating technology can be used to deliver new features and services to customers. Because these can be delivered to users without requiring them to bring their device to a service facility or purchase a new handset, carriers can conveniently deliver new services to individual users’ handsets, in effect “personalizing” the handset to meet the particular needs of an individual user. This has the advantage of enhancing customer loyalty and it provides opportunities for carriers to generate additional revenue per subscriber.

## A Description of the FOTA Update Process

The FOTA update process consists of three primary stages: generating the update, managing the delivery of the update, and performing the update.

### Mobile Devices Susceptible to Flaws and Viruses

- In Japan, several hundred thousand phones from several different vendors were recalled because of software flaws that caused power and email problems.
- In 2004, a worm, labeled Cabir, targeted smartphones, causing display irregularities. The worm was spread to other phones via Bluetooth technology.
- In Spain, a virus spread through email to PCs and crossed onto mobile phones, generating rogue SMS text messages to subscribers.
- In Japan, software flaws exposed phones to viruses sent via text messages. These caused phones to dial the country's emergency services number.

### Generating the Update Package

The initial stage in the process involves generating a software update package containing bug fixes or new features. In order to make this package as small as possible (an issue to be discussed in more detail below), the update package includes only the changes (also referred to as the "delta") between the version that already exists on the device and the updated version. This update package is typically generated by the owner of the software, most often the mobile handset manufacturer.

### Managing the Delivery of the Update Package

Once generated, the update package is published to a distribution platform managed by the carrier or handset manufacturer, depending on who directly manages the relationship with the subscriber. This platform manages the various versions of the update packages and handles the actual networked delivery (download) of the packages to the appropriate handsets. There are typically multiple versions of update packages, each intended for particular handset

models and configurations. This portion of the process can be an integral part of an overall Mobile Device Management (MDM) system.

### Performing the Update

In this third stage of the process, the downloaded update package is used to perform the actual update (re-flashing) of the original software image. The update package and the utility necessary to perform the update occupy a small amount of memory allocated within the handset. (The challenges associated with the limited memory resources are discussed in more detail below.) This stage also validates that the correct update package has been received and that the update process has been successfully completed.



Illustration 1: The OTA update process

## Addressing the Challenges of FOTA Updating

Though FOTA updating can provide significant potential advantages as a mechanism to deliver fixes and new features to mobile handsets, manufacturers and carriers face several challenges to successfully using this process:

- Network bandwidth limitations
- Device memory resource constraints
- Security concerns
- Reliability
- Standards compliance & interoperability

### Network Bandwidth Limitations

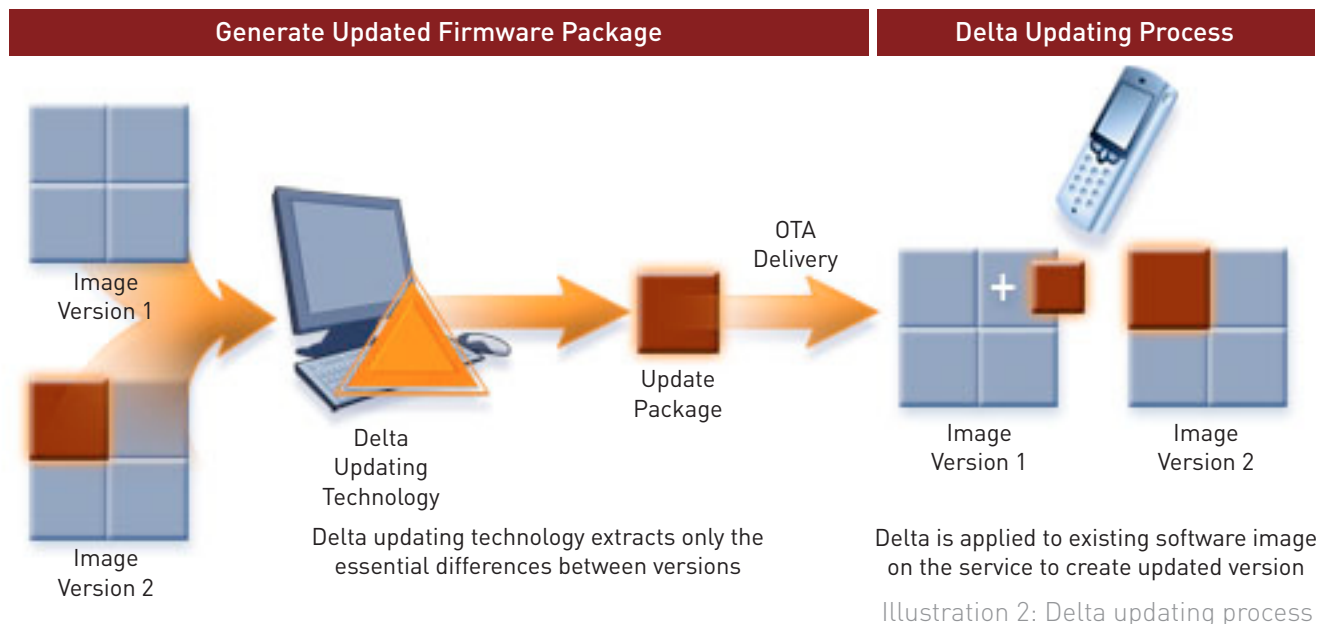
Embedded handset firmware typically requires image sizes in excess of 10 megabytes, and this requirement increases as handsets become more complex. Even with the advent of 3G mobile networks, bandwidth limitations make downloading files of this size difficult. Delivering the complete software image as a replacement update package by a FOTA process would be a lengthy, expensive, inefficient and impractical process. With connection speeds as low as 9.6 kbps, a large update package increases the likelihood of both user frustration and update interruption or failure.

As they evaluate FOTA update solutions, manufacturers and carriers should select an update technology that yields an update package that is significantly smaller than a complete replacement file. As a general rule, only these smaller packages can meet bandwidth, memory and reliability requirements.

To generate these small update packages, carriers and manufacturers should consider the advantages of a technique known as delta updating (See Illustration 2). By contrast with conventional methods, such as compression of the complete updated firmware image, delta updating is significantly more effective in reducing the size of the update package. This advanced technology uses content-aware algorithms that extract only the differences between two software or firmware images. The result is a much smaller software update package, suitable for FOTA delivery. The size of the update package will also have an impact on the time required to perform the update on the mobile handset. In order to minimize the update process duration and reduce user downtime, manufacturers should consider the advantages of faster flash reprogramming as they evaluate FOTA technology.

### Challenges to FOTA Updating

- Network bandwidth limitations
- Device memory resource constraints
- Security concerns
- Reliability
- Standards compliance & interoperability



## Device Memory Resource Constraints

FOTA update package size is constrained by the amount of available memory on the device to perform the update. Most mobile handsets have limited embedded memory. If a large update package is downloaded to the device, there may not be sufficient available memory to store the package and perform the update.

### Memory Constraints

Memory constraints of mobile devices present challenges to FOTA updating:

- Small device size limits space available for memory
- Cost considerations limit memory capacity
- Memory constraints require “in-place updating” of firmware
- Limited memory requires a small footprint for the FOTA update agent resident on the device

To accommodate this condition, embedded software updates require the use of in-place updating — the ability to update device software in place of the currently running version. One of the significant challenges of in-place updating involves the “write-before-read conflict.” FOTA updating technology must overcome this conflict without increasing the update package (delta) size or compromising reliability and process efficiency.

In addition, the FOTA update client agent residing on the handset itself should occupy only a minimal portion of ondevice memory. This agent is required to manage the actual updating process using the update package once it is delivered. Ideally, the on-device element should reside in as

little as one flash sector of on-device memory. Runtime memory must also be used sparingly during the update process. The process should be designed to proceed without fully decompressing the update package into RAM.

Manufacturers and carriers, therefore, must assess the ability of FOTA update technologies to operate within this memory-constrained environment.

### Security

The updating process must be secure and incorruptible, eliminating the risk of unauthorized access to the software operating on the mobile handset. Mobile handsets cannot be exposed to unintended or unauthorized use as a result of an insecure condition introduced through the updating process. To provide the required level of security, the FOTA update solution must provide mechanisms for verification and validation.

Specifically, the mobile handset manufacturers and carriers should consider the advantages of FOTA update solutions that incorporate technology for pre-update validation. This technology is responsible for authenticating a new software version before the update process begins. Using this technology, the FOTA update solution in effect verifies everything related to the process in advance: the delta file, the new version and the process itself, including any previously undetected bugs. The update process will continue only if it can be confirmed that the new version is authorized.

### Reliability

Mobile handset manufacturers and carriers must be assured that the FOTA update process is reliable under all conditions. In many instances, a failed update that disables a handset can be worse than no update at all. Devices cannot be disabled due to errors or unexpected disruptions, even in the extreme range of conditions in which devices are operated.

One critical element of reliability to be considered is how the FOTA update technology handles interruptions in the update process. Mobile handsets are commonly subjected to power loss or other disruptions to normal status. The more advanced FOTA update technologies have provided mechanisms to recognize these interruptions and resume updating when the normal state is restored.

In addition, handset manufacturers should consider the FOTA update technology’s self-update capability. As standards and technology evolve, this self-updating capability will allow the update agent embedded into the handset during the manufacturing process to be upgraded remotely. In other words, the update agent itself can be updated using FOTA technology. This provides the manufacturer a level of continuity, interoperability and investment protection.

## Standards Compliance & Interoperability

To manage their large subscriber base, most carriers have implemented multiple systems to handle network management, billing, provisioning, customer support and other critical functions. These carriers have typically installed different systems from a variety of suppliers to assemble an infrastructure that meets their particular needs.

In considering FOTA update solutions, handset manufacturers and carriers should consider the ability of that solution to integrate with the heterogeneous array of applications installed. The FOTA update technology should comply with industry standards, such as OMA (SyncML)DM, and have demonstrated interoperability with a variety of complementary technologies. Carriers should not put themselves in a position which would require them to replace functioning systems already in place in order to accommodate the FOTA update solution.

## Benefits of Effective FOTA Update Technology

Those manufacturers and carriers that select and implement effective FOTA update solutions can realize significant benefits. The ease with which applications, embedded software, network service clients, and operating systems on customer devices can be updated, upgraded, modified and repaired, can have a direct impact on product success and bottom-line performance. For mobile handset manufacturers and carriers, the ability to quickly, reliably and cost-effectively update software on mobile devices already in the hands of users can be a powerful advantage. It can help them in several ways:

- Accelerate the time-to-market and adoption of new features or services
- Reduce the cost of repair, service and troubleshooting
- Increase ROI and extend product life cycle
- Improve customer satisfaction, strengthen customer loyalty, and boost revenue per subscriber.

## Glossary

**3G networks:** The third generation of wireless networks. These networks must be able to transmit wireless data at 144 kbps at mobile user speeds

**Delta updating:** A sophisticated technique that identifies the essential differences between one version of software and a second version. These differences are also referred to as the “delta” between versions.

**Firmware over-the-air (FOTA) updating:** Refers to the process that allows mobile phone handset manufacturers and carriers to remotely update the software and firmware that controls the functions of the handset.

**Firmware:** Programming instructions that control the core functions of the handset. Used interchangeably with “software” for the purposes of this report. Firmware over-the-air updating is often referred to as FOTA.

**In-place updating:** Refers to a technique for efficiently replacing an existing software version with an updated software version on a mobile device with limited memory.

**Mobile devices:** For the purposes of this report, these refer to a full array of mobile communication instruments including voice-centric phones, enhanced feature phones, and smartphones.

**Mobile device management (MDM):** Refers to the system used by carriers to manage the population of mobile phones in the hands of their subscribers. The system tracks users, devices, configurations, updates, service plans, and other details.

## Red Bend Software's vCurrent® Technology and Products

**vCurrent Technology:** Red Bend's solutions are based on the company's patented vCurrent technology. This technology, proven through the delivery of hundreds of millions of accurate, secure and cost-effective updates, manages key elements of the software update process:

- Identifies the essential changes from an existing version of software to a new updated version
- Generates a new compact package of updated software, up to 97% smaller than can be achieved through conventional compression techniques, to allow for efficient FOTA updating
- Installs the new software within the limited memory available on a remote or wireless device
- Manages different versions of updated software and their deployment on a variety of devices

**vCurrent® Mobile:** vCurrent Mobile enables mobile telephone manufacturers and carriers to create and manage software updates for mobile telephone handsets and to deploy and install the updates over-the-air.

Based on Red Bend's patented technology, vCurrent Mobile is comprised of two primary components: an update generator and an update installer.

- vCurrent Mobile's Update Generator identifies the essential changes from an existing firmware version to a new, updated version and automatically creates an extremely compact package of these changes. The Update Generator also supports updating a handset's file system, which typically stores images, sounds, configuration information, settings, design themes, icons, menus, system status and other information that affects device appearance, configuration and branding.
- vCurrent Mobile's Update Installer resides on the mobile handset itself and performs the update installation. Optimized for the limited memory available within the handset, the vCurrent Mobile Update Installer applies the updates in-place on the phone's firmware accurately and reliably. It performs this update on feature phones with monolithic firmware images and on smart-phones with compressed, multi-section images. The update installer also performs file system updates.

### vCurrent Mobile Advantages

**Proven and patented Red Bend delta-update technology overcomes bandwidth and memory constraints.**

Proven in hundreds of millions of updates, Red Bend's patented delta-update technology extracts only the essential changes between mobile phone firmware versions, reducing update file size by up to 97% and lowering device memory and bandwidth requirements, while maintaining 100% accuracy.

**vCurrent Mobile supports advanced phone architectures.**

Using advanced algorithms, vCurrent Mobile supports Symbian OS and other architectures that employ compressed multi-section images, read-only file systems, a combination of NAND and NOR flash and other advanced features.

**vCurrent Mobile speeds up flash reprogramming.**

By updating only the changed sectors of the firmware image, vCurrent Mobile significantly reduces the time the handset is out of service for updating and minimizes the user's inconvenience. vCurrent Mobile's patented computational update technology can guarantee consistent update performance for an unlimited number of on-device updates, from any firmware version to any other version.

**vCurrent Mobile optimizes on-device memory.**

Using a unique "in place update" algorithm, vCurrent Mobile requires only minimal working memory to perform updates while providing control over the amount of flash memory required. This patented technology resolves complex "write before read" issues so that firmware can be updated rapidly without adding costly memory.

**vCurrent Mobile can be integrated into handset firmware in a matter of days.**

vCurrent Mobile's unique technology and QuickPort™ process allow handset manufacturers to integrate the firmware update technology without restructuring device firmware and with no impact on the firmware development tool chain. As a result, integration has been proven to be significantly faster and less expensive than alternative solutions.

**vCurrent Mobile ensures 100% fault-tolerant update integrity and completion.**

vCurrent Mobile eliminates the risk of disabling devices due to errors or unexpected disruptions in the update process. In the event of a power loss, it automatically resumes the update from the point of interruption.

**vCurrent Mobile ensures security for all stages of the update process.**

Using a patented pre-installation verification technique, vCurrent Mobile software verifies that the newly created software version is 100% accurate, bug-free and will install properly, prior to the update process getting underway.

**vCurrent Mobile operates with other mobile device management solutions.**

vCurrent Mobile works with all existing mobile device data distribution and transfer solutions and all standard protocols.

**vCurrent Mobile Update Installer minimizes the on-device footprint.**

The Update Installer software requires only one flash sector of 64KB as permanent allocation for the Update Installer (about 20KB) and the device specific Update Agent, and one to be temporarily allocated upon running for the update-specific data, to ensure start-to-finish integrity of the update install process.

**vCurrent Mobile Update Installer is update-able over the entire device lifecycle.**

The Update Installer software itself can be quickly updated over the air, allowing it to be embedded during the manufacturing process and upgraded remotely and transparently, as required, throughout the life of the device.

## About Red Bend Software

Red Bend Software helps mobile phone manufacturers and network operators to accelerate the adoption of new services and features, respond rapidly to customer needs, and reduce support costs through mobile software management solutions. LG Electronics, Motorola, NEC, Sharp, Sony Ericsson and other large handset manufacturers use Red Bend's firmware over-the-air (FOTA) mobile client software to quickly and reliably deliver compact firmware updates to more than 150 million mobile phones in the hands of consumers. Founded in 1999, Red Bend Software is a privately held, venture capital-financed company with offices in China, Israel, Japan, Korea, the U.K and the U.S. More information is available at [www.redbend.com](http://www.redbend.com).



400-1 Totten Pond Road, Suite 130  
Waltham, MA 02451, USA

+1-508-270-4590  
+1-508-270-4595 fax

[www.redbend.com](http://www.redbend.com)