



November 2004



**Reliance
Infocomm**

A New Way Of Life

R-EVOLUTION

Newsletter for the Developer Community

For Private Circulation only

Editorial

Dear Developers,

Mobile Commerce is the effective delivery of electronic commerce into the consumer's hand anywhere, using wireless technology. With mobile phone penetration in India already ahead of PCs, the emerging era of mobile commerce promises to permeate all areas of business and private life. Our lead story on Mobile Commerce charts out business opportunities thrown open by this integration of communication and information technologies, both in the consumer and enterprise segments.

In our Application of the Month, we present to you a perspective on RIM Bill, an M-Commerce product well adopted by Reliance customers for its easy to use features and unique benefits.

While Reliance handsets support MIDP 1.0, it does not provide the much-needed mobile media API for developers to creatively package the application. Reliance provides Java extension APIs to meet this requirement. Read more on the Media Package of Reliance API in the new Products and Processes section. In our Technology Tips section, we have provided illustration on Database Connection Pooling in order to minimize customer response time for client-server applications.

As always, we look forward to receiving your valuable suggestions and useful feedback.

Wishing you all a Very Happy Diwali and Prosperous New Year.

- S P Narayanan

M-Commerce - the smart way of doing Transactions



Airline Booking



Mobile Banking



Railway Reservations



RIM Bill



Tech Tip

Database Connectivity



Application of the Month

RIM Bill



New Products & Processes



Interview of the Month

ZMQ Software



Quiz

M-Commerce

- the smart way of doing Transactions

Instant business

The increased mobility and reach of wireless transactions have unleashed enormous opportunities for business enterprises to provide enhanced services to their customers. It makes possible the concept of delivering value to the customer at all times, irrespective of his or her location, as long as he/she is within connection range. The different parties involved in the entire m-commerce value chain are wireless infrastructure providers, wireless service providers, certifying authorities, applications/software providers, equipment manufacturers, credit card companies and banks.

Types of transactions

M-commerce transactions can be generally grouped under two categories - low-value and high-value. Low-value transactions usually imply music downloads, logo downloads, picture downloads, ring tone downloads, etc. There can be some banking and value-added services like balance query and stock alerts. Then there are services like m-coupons and wallets. On the other hand, there are high-value transactions, which involve credit and debit card transactions, point-of-sale terminals, going to the merchant location and paying through the handset. Thus, the customer has the option of paying through credit card, debit card and direct debit. Besides these facilities, a customer can do a transaction on mobile and even pay by cash on delivery.

Global scenario

According to Nasscom's 'Strategic Review 2003', in the year 2002, the m-commerce market was in the region of \$50 billion worldwide. While the US and European markets are expected to dominate forecasted revenues until 2005, Asia-Pacific regions and the rest of the world are expected to account for 40 percent of the estimated \$225 billion m-commerce market by 2005. Revenue for all forms of m-commerce as per Telecom Trends International's prognosis will grow from \$6.86 billion in 2003, to a whopping figure of \$554 billion by 2008.

Interestingly, the worldwide ring tone market was worth \$3.5 billion last year, up 40 percent from 2002 and representing about 10 percent of the global music market, according to estimates from UK-based research firm ARC Group. Most of those sales have taken place in Europe and Asia, where mobile phone markets are more mature. ARC predicts global sales for ringtones will surpass \$5.2 billion by 2008. It is widely believed that the growth of micro payments will drive goods and services over mobile phones. According to Juniper Research, retailers are expected to ring up \$40 billion worldwide in sales from customers who will use mobile phones to spend cash on everything from cinema tickets to hourly car parking. The figures exclude the already popular m-commerce products like mobile phone ringtones, Java games, Wall Papers and screensavers.

India's progress

M-Commerce is still at its nascent stage in India. However, it has immense potential in view of the enormous growth of mobile phones in this country. Fierce competition among operators has consistently driven down tariffs, reducing revenues from the voice-based operations of the wireless networks. The advent of wireless Internet in the form of CDMA 1x 2000 will boost m-commerce in India. Analysts predict that the growth rate in the value-added services, of which M-commerce is a part, in India could be in the range of 15 per cent over the next five years.

Reliance Infocomm's rapid strides

Reliance Infocomm has taken a lead in India in providing the necessary infrastructure and technology for the growth and development of m-commerce in the country. A number of applications such as mobile banking, railway booking and RIM bill payment have already been launched in its R World suite. Many such transaction-oriented applications are also in the pipeline.

Not only are these applications easy to access and extremely secure, but offer users with an unbeatable cost advantage compared to similar services internationally provided by operators. For example, in Japan a NTT

DoCoMo subscriber has to pay a fixed monthly fee of 315 Yen (1 US \$ = 105.47 Japanese Yen ; 1 Japanese Yen = 0.43 INR) which comprises a 157.5 Yen packet transmission charge and a 157.5 Yen fee for i-mode service. Other services range from 7.35 Yen for downloading still images, to 27.3 Yen for checking share prices, to 63 Yen for transferring funds; for RIM users the same /similar services are currently offered for free of cost under R World.

“Reliance early initiatives in m-commerce services have resulted in over 400,000 RIM subscribers registering for m-Commerce. Launch of RIM Bill Pay, Railway booking and Mobile Banking has met with enthusiastic response from



Anil Pande

RIM subscribers. Reliance shall leverage its technological capabilities and its relationships to shortly launch a slew of m-Commerce applications such as Shopping, Airline Ticketing, Auctions, Movie and Event ticketing etc. So lets all be ready for lots of exciting action in the M-commerce space,” gushes Anil Pande, Head Enterprise Solutions, ASG, and Reliance Infocomm.

One of the major concerns for successful implementation of any transaction-based application is security, which is why while designing any such application; Reliance takes care to ensure that these issues are addressed to ensure end-to-end application security. While the RIM Bill pay application currently uses the 128-bit SSL encryption, which makes data transfer totally foolproof and safe, applications like Railway reservation fully comply with security concerns like Data Privacy, Authentication, Message Integrity, Non-Repudiation and Access Control.

Market players

The m-commerce market players are classified into three types: network operators, suppliers and content providers. Each of them has resources that can diversify into the field of m-commerce. Network operators have an established customer base and control of the networks. For them, m-commerce is a way to build stronger customer relationships and generate useful customer data. Suppliers include handset and network equipment manufacturers, application system developers as well as security specialists. Content providers include

companies that offer wallpapers/ringtones etc. As well as merchant establishments selling services/merchandise.

Potential applications & opportunities

The move towards m-commerce is gathering pace. M-commerce applications are already allowing users to purchase products, make airline reservations, and check bank balances over cell phones, as well as access real time stock quotes and personal financial information. In the next few years, the number of applications will considerably grow and greatly benefit finance, travel, hospitality, retail and entertainment sectors. The sales and usage of mobile phones are soaring, outpacing existing and forecast PC sales. M-commerce could prove to be a very promising opportunity for cellular operators / application developers / merchant establishments to expand subscriber numbers, increase revenues, increase customer loyalty, and differentiate from their competition.

M-commerce has many advantages in today's market. Presently, many companies are moving towards a one-to-one customer marketing relationship where a seller positions himself in order to reach a single buyer unlike PCs and televisions. As a seller, mobile technology cannot only enable a one-to-one marketing strategy based on specific and individual demographic information but also permit a relationship to exist any time and anywhere. The greatest benefit for the consumer will be the new experience created by m-commerce. Consumers will be able to access desired products anytime and anywhere. Thus, m-commerce can more easily bring the buyer and seller together and facilitate greater profits and a closer customer relationship.

Future trends

Major future trends in m-commerce include transition of commerce on PCs to carry out transactions on mobile devices - leading to better web browsing functionality, move toward third generation cellular technology to deal with compatibility and infrastructure issues, personalization of wireless devices based on location and context, leading to the need for security and changes in marketing and connectivity of major electronic devices. However, the key factors for m-commerce growth will be simplicity and security. Ultimately, the success or failure of mobile commerce, either macro or micro payments, will boil down to these.

Database Connectivity using JNDI

The emergence of data services on mobile devices has added a new dimension to the design of applications. More and more applications are thought of and rolled out every day in some corner of the globe. Even environments that are hitherto sensitive to security have started opening up for mobile data services. With the increase in penetration of mobiles in today's world, the usage of these applications has also tremendously gone up. Also, advancements in technology have heightened the expectations of the customers.

One of the important parameters of customer satisfaction is the response time of any application on his/her handset. With more and more database applications coming up, database handling can be a bottleneck for the overall response of the application. Each time a resource attempts to access a database, it must connect to that database. A database connection involves overhead - it requires resources to create the connection, maintain it, and then release it when it is no longer required. Opening a connection to a database is a time-consuming process. For short queries, it can take much longer time to open the connection than to perform the actual database retrieval. The overhead is particularly high for mobile applications as the storage capacities in these devices are very limited. Consequently, it makes sense to reuse connection objects in applications that repeatedly connect to the same database. This phenomenon is known as Connection Pooling. Most of the current application servers have Connection Pooling in-built in them. The application service provider implements the pool manager in its memory space and can optimize resource usage by dynamically altering the pool size, based on demand.

Connection Pooling is the maintenance of a group of database connections for reuse by applications on an application server. It is part of the JDBC 2.0 Optional Package API. This API provides DataSource, a new means of obtaining connections to a database. The benefit of using the DataSource model is that the creation and management of the connection factory is centralized. Applications do not need to have specific information such as the database name, user name, or password in order to obtain a connection to the database. A DataSource object is retrieved from the JNDI naming service and a connection object is obtained from it. This can then be used as a normal connection object for database operations. Once the requirement is over, this object is then closed as a normal connection. The connection returns to the pool for later reuse. The implementation of the same is given in list-1. The sample properties file is given in list 2.

It is mandatory for any application that is to be deployed in Reliance environment to implement the database connectivity in this manner using JNDI. Hence, DADP Developers are requested to follow the same.

List - 1

```
private Connection con = null;
private PropertyResourceBundle bundle = (PropertyResourceBundle)
    PropertyResourceBundle.getBundle("phone");
private DataSource ds = null;
private Logger logger = new Logger();

String dburl_str = bundle.getString("DBURL");
String dbdriver_str = bundle.getString("DBDRIVER");
String dbuser_str = bundle.getString("DBUSER");
String dbpasswd_str = bundle.getString("DBPASSWD");
String dbpl_source = bundle.getString("DB_POOL_NAME");
String connectionType = bundle.getString("CONNECTION_TYPE");

public Connection openConnection(boolean setAutoCommit)
try {
    logger.log(this.getClass().getName(), "Request to open a new connection", "");
    if(connectionType.equals("1")) {
        Class.forName(dbdriver_str).newInstance();
        con = DriverManager.getConnection(dburl_str, dbuser_str, dbpasswd_str);
    } else {
        if(ds == null) {
            logger.log(this.getClass().getName(), "=> DataSource creating new..", "");
            InitialContext initialCtx = getContext();
            ds = (javax.sql.DataSource) initialCtx.lookup(dbpl_source);
        }
        con = ds.getConnection();
        if (con != null) {
            logger.log(this.getClass().getName(), "Getting Connection String");
        }
    }
} catch (ClassNotFoundException cnfe) {
    logger.log("CNFEXCEPTION : openConnection() => " + cnfe.toString());
} catch (SQLException sqle) {
    logger.log("SQLEXCEPTION : openConnection() => " + sqle.toString());
} catch (NamingException nex){
    logger.log("NAMINGEXCEPTION : openConnection() => " + nex.toString());
} catch (Exception ex) {
    logger.log("EXCEPTION : openConnection() => " + ex.toString());
}
return con;
}

public void closeConnection(Connection connection)
try {
    if (connection != null)
        connection.close();
} catch (SQLException e) {
    logger.log(this.getClass().getName(), "closeConnection" ,e.getMessage());
} catch (Exception e) {
    logger.log(this.getClass().getName(), "closeConnection" ,e.getMessage());
} finally {
    connection.close();
    connection = null;
}
}
```

List - 2

```
#ORACLE DATABASE Data Source Details
# DATA SOURCE CONNECTION_TYPE = 2
# JDBC CONNECTION CONNECTION_TYPE = 1

CONNECTION_TYPE=1
DBURL=jdbc:oracle:thin:@10.8.149.168:1521:APPS
DB_POOL_NAME=jdbc/ GSMPHONE

#ORACLE DATABASE Connection Details LIKE DBNAME USER AND PASSWORD
DBDRIVER=oracle.jdbc.driver.OracleDriver
DBUSER=GSMPHONE
DBPASSWD=GSMPHONE
```

Click and pay

Over the past few months, Reliance has devised a number of customer-friendly and secure m-commerce applications that are both time saving and hassle free. In this issue, we present the salient features of one such very popular and useful application called RIM Bill.

Convenient Method

The RIM Bill is a suite of applications, which allows RIM subscribers to view, pay and check other usage details through their mobile itself - anytime, anywhere.

Using the **RIM Bill** application, users can pay their phone bill through various payment options. This includes payment through credit cards (VISA/MASTER/DINERS) or through direct debit facility from their HDFC or ICICI Bank Accounts.

To pay by credit cards, subscribers need to do a one-time registration with Reliance Infocomm and obtain a four digit secure PIN through the Get PIN application in the same category. To start paying by Direct Debit through HDFC or ICICI Bank Accounts, the subscribers need to register with the respective bank for Mobile Banking and Bill Payment facility. They also need to obtain the four digit secure PIN through the Get PIN application.

The Application also has the provision to help the user view the status and details of the Last Transaction. This is very useful in the event of a network glitch at the time of processing the payment request. Users can also use this as proof of payment.

Through the **View Bill** application, subscribers can easily view their bill details on their RIM phone at the click of a button. They can view details such as STD calls, ISD calls, SMS, monthly charges, Carry Forward minutes etc, for the current as well as previous months.

A new feature is being added in View Bill application wherein subscribers would be able to request for a copy of

their bill (current or previous) to be sent to an email address, which they may enter in the same application.

Highly secure

The RIM Bill application is completely secure. Subscribers can use the application only if they are registered with Reliance Infocomm and know their four-digit PIN. Hence, each subscriber can pay one's bill only through his/her handset. The PIN entered by the user is hashed in the handset itself and then sent over the air to the server. Also, the PIN is hashed and stored in the PIN Management System, making it completely secure. The credit card payment request is encrypted using 128-bit SSL encryption between the handset and the RAP server and also between the RAP server and the bank, thereby ensuring end-to-end security. This makes sure that no one can view the payment request details. Also, the RAP servers and the payment gateway servers use a digital certificate for authentication. In addition, a checksum key validation is done for each request response to make sure that there is no corruption of data.

Using the **Unbill Usage** application, a RIM subscriber can find out his unbilled usage. This is the total amount payable by him from the date on which his last bill was generated to the current date. This helps subscribers track their usage on a day-to-day basis.

Subscribers can also see the last payment made by them. They can see the amount they have paid and the date on which the payment was received. Thus, subscribers can find out if the payment made by them has been received by Reliance.

In the coming days, Reliance plans to launch extensions to this highly popular application, like **Payment Information**, which will provide details related to the previous payments made by a subscriber such as payment date, amount, status of the payment - whether it is accepted, rejected or under processing, location where payment was made, mode of payment, details of payment instrument, etc.

"Currently, customers can pay their RIM bills through credit card and direct debit facilities of ICICI and HDFC banks. Very soon, we shall enable the same for other banks. These features will add to customer convenience and bring forth the power of mobility in the true sense of the term," explains Pooja Rahwani of the Enterprise Solutions Division.



Pooja Rahwani

Media Package of Reliance API

The DADP Team constantly receives queries from developers on Reliance APIs.

These APIs are classes written on top of MIDP 1.0. Most of these APIs are burnt on the Reliance Handsets and can be used in the application. The remainder is kept in the JAR file of the specific application.

MIDP 1.0 has some limitations like not being able to support audio etc. Reliance APIs are a set of classes built on top of MIDP 1.0 to provide extra features and functionality to the applications. To solve problems of the DADP Developer Community, the team is coming up with a series of articles to on these APIs. Provided below are details on the most sought after API package, namely the Media Package. This package is of great importance to many applications having audio content. This mainly involves gaming and ring tone applications.

The Media Package primarily has a class named *Manager*, four interfaces named *Player*, *Control*, *Controllable* and *PlayerListener* and an exception class named *MediaException*. *Manager* is the access point for obtaining system dependent resources such as *Players* for multimedia processing. *Manager* provides access to an implementation specific mechanism for constructing *Players*. *Manager* provides the method *createPlayer* to create a player for playing back media from a byte collection, which returns the object of the player. The player returned can be used to control the presentation of the media.

Player is a subclass of the super interface *Controllable*. It controls the rendering of time based media data. It provides the methods to manage the players' life cycle, controls the playback progress, obtains the presentation components, controls and provides the means to synchronize with other players. After the player is created from *Manager*, calling start will commence the playback as soon as possible. The start method asynchronously starts the player - i.e., the method will return as soon as the playback has started. The playback will continue in the background and by default, automatically stops when it reaches the end of the media. A player has four states: UNREALIZED, REALIZED, PRE-FETCHED, and STARTED. The purpose of these life cycles is to provide programmed control over potentially time-consuming operations. Typically, a player moves from the UNREALIZED state to the REALIZED state, then to the PRE-FETCHED state, and finally on to the STARTED state. When the player stops, it moves from the STARTED state back to the PRE-FETCHED or possibly back to the REALIZED state. It is then ready to repeat the cycle.

Controllable provides an interface for obtaining the controls from an object like a player. It provides methods to query all the supported controls and to obtain a particular control, based on its class name.

PlayerListener is the interface for delivering asynchronous events generated by *PlayerUpdate*. It has a method called *Player Update*, to deliver an event to a registered listener when a player event is observed. It will only be called by the native player mechanism whenever relevant events arrive from the CODEC. Media Exception indicates an unexpected error condition in a method.

Along with the above-mentioned classes, Media Package also has a sub package called *media.control* with *VolumeControl* and *ToneControl* classes, which implement Interface control. *Volume Control* is a class for manipulating audio volume. This class allows the output volume to be specified using an integer value that varies between 0 and 100. *Tone Control* is a class meant to enable playback of a user-defined monotonic tone sequence. A tone sequence is specified as a list of tone-duration pairs. The list is packaged as an array of integers. The set sequence method can be used to input the sequence to the *Tone Control*. A sample implementation is given in the list shown alongside.

List - 1

```
public void loadPlayer()
{
    InputStream stream1 = null;
    byte [] ba = new byte[4500];
    Player player = null;
    try
    {
        stream = this.getClass().getResourceAsStream("party2.mid");
        int result = 0;
        int bigRes = 0;
        while (result!=1)
        {
            result =stream.read(ba,bigRes,4500 - bigRes);
            bigRes+=result;
        }
        bigRes++;
        byte[] newBa = new byte [bigRes];
        player = Manager.createPlayer(newBa, null);
        player.start();
    }catch(MediaException me)
    {
        me.printStackTrace();
    }
}
```



'Right kind of game logic is the key to developing innovative and interesting games'

New Delhi based ZMQ Software Systems is an Edutainment company specializing in gaming and learning, headed by Subhi Quraishi, M.Tech, its CEO. A workaholic by nature, Subhi spends nearly fourteen hours on concept, designing, planning and meticulous execution, along with his twin brother Hilmi Quraishi. Turning ideas into innovative and interactive solutions are Subhi's avowed mission. We bring you Subhi in our Interview of the Month.

Could you tell us about your experience as a Developer working for Reliance?

It is a great opportunity to be working with Reliance Infocomm, India's largest mobile service provider with over eight million customers. The Technical Support Team of DADP with whom we are working has been very co-operative from day one.

What prompted you to develop the India XI game?

India XI Safari is the first in the series of adventure based mobile games made on Indian Cricket heroes. Cricket is like a religion in India and cricketers are treated like demigods. Our aim is to develop something adventurous and innovative around cricket where users can play and interact with their favorite cricketers.

How is your game different from similar other games?

The right kind of game logic and sound instructional design are the key elements to developing innovative and interesting games. We have a team dedicated to analyzing the psychology and new excitement components of gaming. Plus, we have a specialized team engaged in mobile handset instructional design. Games developed at ZMQ are original. We keep in view the target audience. Technically, mobile games designed at ZMQ are light in weight (small memory foot print) and are highly portable for a wide variety of devices. ZMQ has immense experience in designing high quality scripted animations and graphics, which make our games highly popular.

What are the future extensions to the India XI game?

The India XI safari is being adapted as a multi-level game with increased layer of difficulties. We are adding five new characters called 'The Big Five of Safari', which will chase the Zombie and make his task more challenging.

What other games have you developed or are developing for Reliance?

ZMQ has already developed three games for R-World out of which one (India XI) has already gone live. The other two games are Micrtotrix - a mobile arcade game and The Great Escape - an adventure game. The beauty of The Great Escape game is its simple maneuverability as it can be played anywhere. Both the games are undergoing tests



Subhi Quraishi

with the DADP Team. We are also coming up with more games and valuable m-edutainment and m-enterprise solutions for the Reliance Network.

Any interesting or memorable event that you would like to share with us about your application development work with DADP?

Game development is an exciting process. During the course of development of our games, there have been many interesting moments while interacting with the DADP Team. The most memorable of course was when we learnt from my friend that India XI Safari game was on the R-World, on the first day of its launch. It was a proud moment for all of us.

Your comments on the mobile gaming/application scenario in India today and the future trends.

The mobile is the device of the future. Whether it is communication, entertainment, gaming, education or any other service, mobile devices will hold the key to all. As these devices gain in strength, there will be simultaneous gain in scope. Gaming and other application will grow, but for the Indian scenario we must focus on Indian ethnic content for making it more effective. House of Learning, the Social Initiative Program of ZMQ Software Systems, is engaged in developing high value mobile learning solutions focused on the semi-literates and the underprivileged of India. We believe that service providers and organizations like Reliance Infocomm can really make a difference with their wide spread network to influence a social change by creating awareness in the masses.

Dear Sir,

Thank you once again for this great edition of your newsletter. I want to know about the universities that you are covering in your university relationships program. Also, are there any universities from Delhi? The launch of the WAP OTA tool is a great resource for developers who have waited long before they could test their applications on the handset itself. I want to activate my number for it. Please let me know how to do it. Thanks.

Rahul. (Developer)

Hi Team,

I really liked the cover story of your previous issue on URP. I'd like you to extend this program to other cities and towns as well just like the RIM network. Being an academican I fully understand the importance of such a program of student-industry interaction, hence the suggestion to expand the domains of the program.

Keep up the great work!

Regards,

Samuel K. (Thiruvananthapuram)

Editor's note: Thanks for your email. Currently, the University Relations Programme has been launched on a pilot basis in select Engineering colleges in Mumbai and in DAICT, Gujarat. In future, we plan to extend this to cover other colleges/ Universities across the country in a phased manner.

To activate your number for the WAP OTA tool, please send an email to dadp.query@relianceinfo.com giving your mobile number with STD code and the model name of the handset.

Quiz

1. Which of the following is not a required field of a JAD file.

- | | |
|-------------------|--------------------|
| A) MIDlet-Name | C) MIDletData Size |
| B) MIDlet-Jar-URL | D) MIDlet-Version |

2. When executed, the following line of code will print

`System.out.println (-1 * Double.NEGATIVE_INFINITY);`

- | | |
|--------------|---------|
| A) -Infinity | C) NaN |
| B) Infinity | D) -NaN |

3. For a class defined inside a method, what rule governs access to the variables of the enclosing method?

- | | |
|---|--|
| A) The class can access any variable | C) The class can only access transient variables |
| B) The class can only access static variables | D) The class can only access final variables |

4. Which of the following is the correct syntax for suggesting that the JVM performs garbage collection?

- | | |
|---|---------------------------------|
| A) <code>System.free()</code> | C) <code>System.out.gc()</code> |
| B) <code>System.setGarbageCollection()</code> | D) <code>System.gc()</code> |

5. ARPU in telecom terminology stands for:

- | | |
|-----------------------------|-----------------------------|
| A) Average Revenue Per User | C) Amount Received Per User |
| B) Average Revenue Per Use | D) None of the above |

Answers : 1. C) 2. B) 3. D) 4. D) 5. A)



Monthly Poser

In 1998 Sun released a version of Java for to be used on mobile and other resource constrained devices.

Which of the following is it?

- A) Personal Java
- B) J2ME
- C) Java on Mobile
- D) MicroJava.

Answer to last month's poser: GREEN TEAM

Last month's winner of the Monthly Poser through a Lucky Draw is: Pandurang Nayak, Bangalore.

Winner will receive his prize by courier.

Answers to the Monthly Poser should be sent to dadp.newsletter@relianceinfo.com mentioning Monthly Poser - November 2004 as the subject. The winner will be decided on the basis of a lucky draw and walk away with a prize from DADP.

You can contribute ideas and information to R-evolution at the following e-mail address: dadp.newsletter@relianceinfo.com.

Please note that contributions may be edited for clarity, style or length.

Editorial Team : S P Narayanan, Saurabh Chakrabarti, Amit Behl, Amit Chandra, Suresh Dabbara, Nitin Saxena, Vikram Vishwanath, Lakshman S Aiyar, Vijayam Raghunathan

Design Team: Makrand Pange, Jaydeep Gholap

Published by the DADP and Corporate Communications Teams, Reliance Infocomm, DAKC, Navi Mumbai