



April 2006

# R-EVOLUTION

Newsletter for the Developer Community

For Private Circulation only



## Editorial

Dear Developers:

The results of Reliance's second mobile application contest have been declared. In keeping with the theme of innovation and practical usage of applications in mind, our jury picked up eight winners: three from general category and five from student category for the top honours.

It was a close call for many of the winners as the overall number and quality of submissions in this year's contest was very high. The jury panel had a tough time choosing the winners. In spite of limited resources and skills at their disposal, the student developers exhibited great deal of talent and developed some very outstanding applications in the contest that caught the attention of the judges who specially emphasised this point. This certainly augurs well for the continuation of our University Relations Programme with more zeal.

You can read more about our eminent jury members, the winning applications, and revenue earning opportunities available to prize-winning developers through commercial launch of their applications on Reliance network in our lead story in this issue.

Our guest article continues with the second part of the 'Creating UI with canvas in J2ME' written by Ashwin – a Reliance Developer Programme member from Hyderabad, while the 'Tech Tip' section talks about XML parsing in J2ME.

Watch out for more exciting announcements for developers in R-evolution in the coming issues.

**Saurabh Chakrabarti**

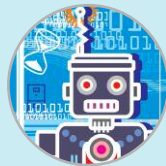
Reliance Developer Programme team



**LEAD STORY**  
Contest Winners  
Page 2



**TECHTIP**  
J2ME  
Page 3



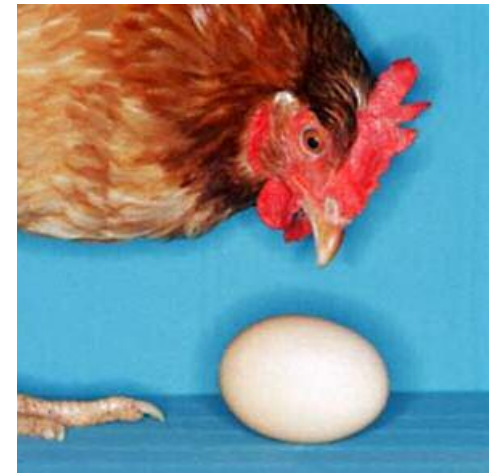
**GUEST COLUMN**  
Creating Canvas  
Page 5



**FEEDBACK**  
Letters to Editor  
Page 7

## CELEBRATING OUR CONTEST WINNERS

Page 2



By making educational institutes a part of this incubation process, we believe that it would foster product development and entrepreneurial spirit

# CELEBRATING OUR CONTEST WINNERS



## The entries

A total of eight entries — three from the general category and five from student category—were chosen winners in this year's contest by a panel of external jury comprising Dr Vijay Mukhi - Cyber Security Expert, Datta Subrahmanya from CISCO, and Prashant Dogra from Forum Nokia. The prize-winning applications include 'Hanuman Chalisa' on mobile, Mobile Blogger, Language tutorial application, animated greetings applications, and a couple of interesting games.

## Revenue share

What is significant this year, apart from the prize money and recognition that winners and participants will get, is the opportunity to make money for their applications through a revenue share arrangement with Reliance Infocomm after commercial launch of these applications on the Reliance network.

## Earn while you learn

To enable budding entrepreneurs of our student community to monetise their innovation, Reliance Infocomm Ltd.(RIC) has formalised a tripartite Revenue Share Agreement between RIC, students, and the respective Institutes as mobile subscribers use their applications. By making educational institutes a part of this incubation process, we believe that it would foster product development and entrepreneurial spirit in the country, especially among the student community.



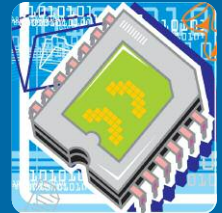
Evaluation of entries of the contest at DAKC

## Grow the community

With initiatives like Mobile Application Contest and University Relationship Program, with support from industry bodies like NASSCOM, the Reliance Developer Program aims to provide continuous support and impart recognition to the developer community. The objective is to share knowledge and establish commercial relations with the developers including established companies, individual entrepreneurs, budding university students and business professionals. The aim is to grow the market and introduce innovative mobile products and services and offer revenue generating opportunities for developers.

For more details on the contest and the winners, visit: [www.dadp.com](http://www.dadp.com)

Reliance Infocomm's Second Mobile Application Contest 2006, along the lines of the first contest held in 2004-05 in association with NASSCOM, received an overwhelming response from the developer community. Over 60 per cent of the total submissions in this year's contest have been from the student community through the University Relationship Program that was kicked off last year with several universities/colleges across India. This year the contest was a big success in terms of the number of entries. We received a total of 52 submissions which saw high levels of innovation. The entries were related to various topics like devotion, fun & entertainment, games, music, etc.



TECH TIP

# XML PARSING WITH J2ME – PART I

With advancement in technology, more enterprise applications are making use of XML (Extended Markup language) as a medium to store data. But mobile applications have largely been left out of this trend due to the increased processing power required by XML parsers. However during the last couple of years there have been applications that have witnessed footprints of XML. The coming together of java and XML creates a powerful combi-

nation of portable data and code.

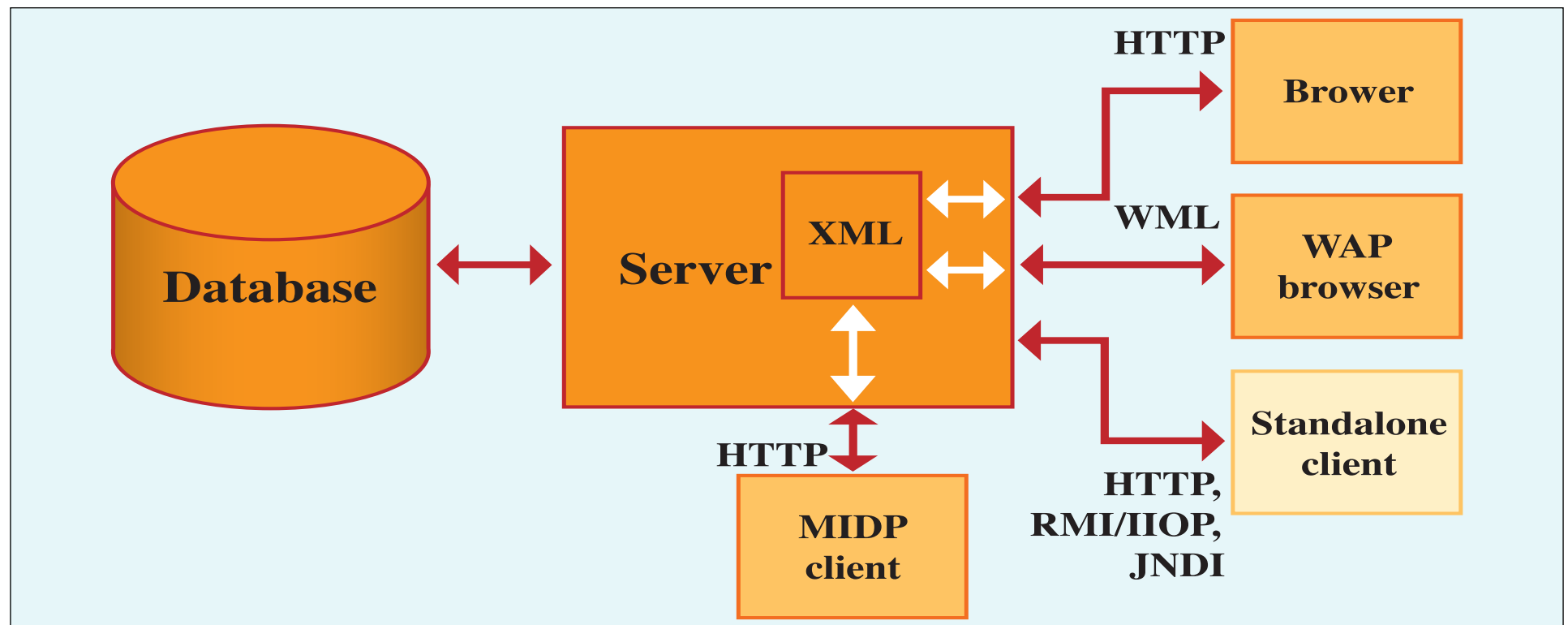
The figure on this page shows an implementation of a multi-tier system that supports HTML browsers, WAP browsers, standalone clients and MIDP clients.

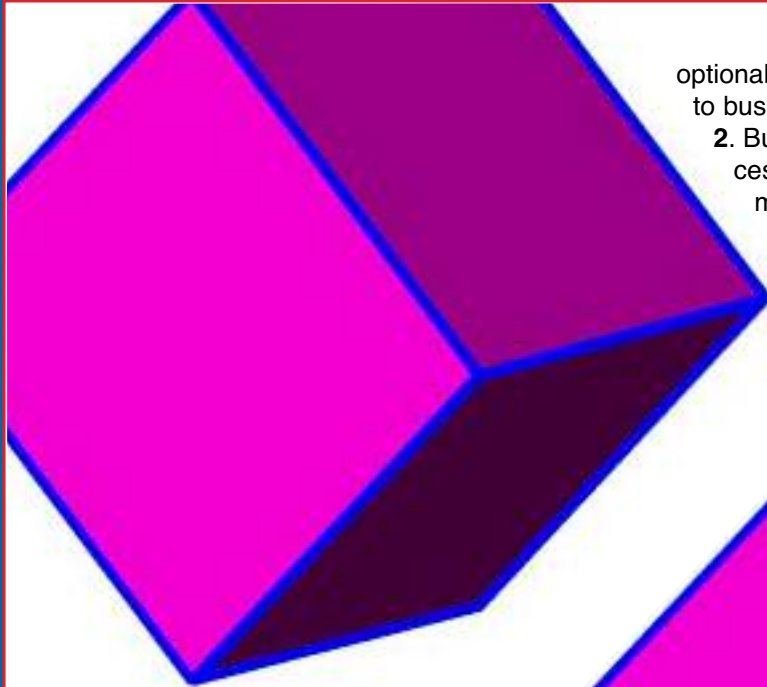
## XML parsers

An XML parser is the software that reads XML files and makes the information from those files available

to applications and programming languages, usually through a known interface like the Document Object Model (DOM). The XML parser is responsible for testing whether a document is well formed and if given a DTD or XML schema, it will also check for validity to determine if the document follows the rules of the DTD or schema.

In simpler words, XML processing is a series of

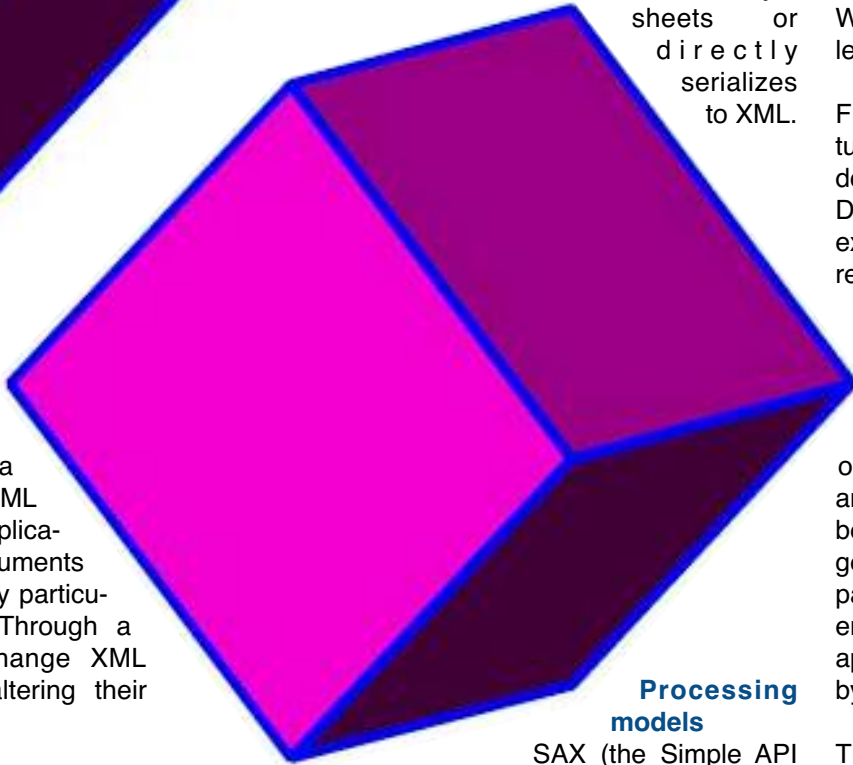




optionally, maps and binds the retrieved information to business objects.

2. Business logic handling - Here, the actual processing of the input information takes place. It might result in the generation of output information.

3. XML output processing - The application constructs a model of the document to be generated with the DOM. It then either applies XSLT style sheets or directly serializes to XML.



steps that an application should take in order to process an XML; an application that implements such a model is called an XML parser. You can integrate an XML parser into Java applications with the Java API for XML Processing (JAXP). JAXP allows applications to parse and transform XML documents using an API that is independent of any particular XML processor implementation. Through a plug-in scheme, developers can change XML processor implementations without altering their applications.

### Three phases

The XML parsing process operates in three phases:

1. XML input processing - The application parses and validates the source document; recognises and searches for relevant information based on its location or its tagging in the source document; extracts the relevant information when it is located; and,

most common processing models. If you use a SAX-based parser to process an XML document, you need to code methods to handle events generated by the parser as it encounters the different tokens of the markup language. SAX-based parsers fall into the category of push parsers.

### Processing models

A push parser reads through an entire XML document. As it encounters various parts of the document, it notifies a listener object.

If you use a DOM-based parser, you need to write a code to walk through the tree-like data structure that the parser will create from the source document. With DOM, the XML input processing is done in at least two cycles.

First, the DOM parser creates a tree-like data structure, called a DOM tree that models the XML source document; then the application walks through the DOM tree, searching for relevant information to extract and further process. This last cycle can be repeated as many times as necessary, as the DOM tree persists in memory. DOM-based parsers fall into the category of model parsers.

A model parser reads an entire document and creates a representation of the document in memory. Both push and model parsers require an amount of memory and processing power that is beyond the capabilities of many J2ME devices. To get around those device limitations, a third type of parser, called a pull parser, can be used. A pull parser reads a small amount of a document at once. The application drives the parser through the document by repeatedly requesting the next piece.

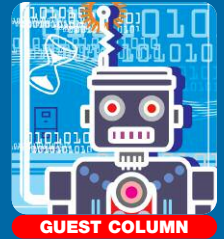
The XML parser to be used in the sample application (an open source application that can be downloaded from the Internet) is an example of a pull parser.

*(To be continued in the next issue)*

For clarifications write to [dadp.newsletter@reliance-info.com](mailto:dadp.newsletter@reliance-info.com)

# CREATING UI WITH CANVAS

## IN J2ME – PART II



In the previous issue we listed all the features of creating Lists. Let us start codifying what we wish to provide to users in our Lists.

To begin, let us create a sample interface, which gives an overview of what we will provide to an application developer to use.

```
public interface AlphaUIInterface extends CommandListener{

    public void setString(String[] aStrArray);
    public void setSelectedIndex(int i);
    public void setSelectedString(String lstr);
    public String getString(int i);
    public String getSelectedString();
    public void setTitle(String aStrTitle);
    public void setAlphabeticScrolling(boolean bool);
    public void commandAction(Command c, Displayable d);
}
```

We have now created an interface so that we can have multiple implementations of List and the implementations can be hidden from the developer.

### Four-way scrolling

Scrolling left, right, up, and down requires us to handle game actions of the phone. So we need to handle events in a keypressed method. This is how it would look:



```

protected void keyPressed(int keyCode) {
    switch (getGameAction(keyCode))
    {
    case UP:
        <!--Place your logic here-->
        break;
    case DOWN:
        <!--Place your logic here-->
        break;
    case LEFT:
        <!--Place your logic here-->
        break;
    case RIGHT:
        <!--Place your logic here-->
        break;
    case FIRE:
        <!--Place your logic here-->
        break;
    }

    repaint();
}

```

The above code is self-explanatory. Don't forget that you need to call `repaint()` after every action as you have to reflect changes in parameters on the display.

### Show title

A Title is a string we draw on the top of the screen shown to the user. I'm storing the title in a String called 'cStrTitle'. For a naming convention I prefixed all class variables with 'c' and type of variable. For example, for the title, as title is a String, I named it `c+Str+Title`.

You need to do the following in the `setTitle` method.

```

public void setTitle(String aStrTitle){
    cStrTitle = aStrTitle;
}

```

### Show Scrollbar (vertical and horizontal)

To show a vertical bar we find a present index of selected string and divide it with the total number of elements in the list to find out the position in which we have to draw a bar. Two rectangular bars are drawn on the extreme right of the screen. One is drawn inside another with the inner one representing the present position in the list.

```

g.setColor(Utils.black);
// Vertical Scroll Bar
g.drawRect(x1,y1,x2,y2);// Outer rectangle
g.fillRect(x3,y3,x4,y4);// Inner rectangle

```

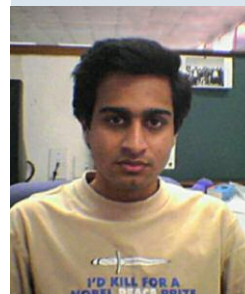
### Index List

We need to index the list so that we can get the contents by name and position. This will help track the positions of different items in the list. This is accomplished in the `getString` method of the list. It also helps to return user Strings corresponding to queries like:

GetString at specified position.  
GetString with given name etc.

We can also perform the task without indexing the list and by basic principle indexing list. This would give you better performance rather than looking for them at runtime which adds extra load to the small memory and processing power of the devices.

This completes the basic tutorial on building UI framework using Canvas. I'll come back again with another tutorial soon.



### About the author

**Ashwin Kumar** is an active Reliance Developer Programme (RDP) member currently working in a major IT company at Hyderabad. He is passionate about Java and has developed a number of mobile applications based on J2ME. He has also participated in many RDP application development contests.



QUIZ

# BRAINGYM

**1. Which of the following audio playback formats must MMAPI 1.1 support in a JTWI compliant device?**

- a) MP3 format
- b) MIDI format
- c) WAV format
- d) Tone sequence format

**2. Which one of the following characteristics defines an untrusted MIDlet suite?**

- a) Access is made by the MIDlet suite to security protected resources or APIs
- b) Origin and integrity of MIDlet suite JAR file cannot be determined
- c) The JAR file origin and integrity can be successfully validated
- d) MIDlet suite fails installation

**3. The MIDP 2.0 Push Registry handles which of the following events?**

- a) Persistent record store events
- b) Incoming network connections
- c) Periodical MIDlet scheduling
- d) User-interface events
- e) Alarm-based MIDlet launch

**4. Which of the following features are extensions of MMAPI 1.1 to the MIDP 2.0 media support through the javax.microedition.media package?**

- a) Sound playback
- b) HTTP 1.1 media transfer
- c) Tone generation
- d) Definition of custom media transfer protocols using DataSource
- e) Video playback

**Answers:** 1b and d, 2b, 3b and e, 4d and e

## Monthly poser

What was born from the 'Operation Somerset' initiated by IBM, Apple, and Motorola?

- 1. Power Play
- 2. Power Mac
- 3. Power PC
- 4. None of the above

**Answer to last month's poser: Chai**

The winner of last month's poser (through a lucky draw) is **Indu Roy - Kharagpur, West Bengal**. Winner will receive prize by courier.

**Answers to the Monthly Poser should be sent to [dadp.newsletter@relianceinfo.com](mailto:dadp.newsletter@relianceinfo.com) mentioning Monthly Poser—April 2006 as the subject with the sender's location stated. The winner will be decided on the basis of a lucky draw and walk away with a prize.**

You can contribute ideas and information to R-evolution at the following e-mail address: [dadp.newsletter@relianceinfo.com](mailto:dadp.newsletter@relianceinfo.com). Please note that contributions may be edited for clarity, style or length.

Editorial Team : Saurabh Chakrabarti, Abhijeet Anand, Lakshman S Aiyar, Vijayam Raghunathan

Design: Ritu Nanda, Haresh Patel, Anand Sutar, Rohidas Adavkar

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



FEEDBACK

# Letters to Editor

**W**e have a few ideas regarding mobile applications like games, entertainment, and related services for young adults. How do we share it with Reliance?

**Ajay Pathak, Sameer Sharma - Chandigarh**

**Editor:** Dear Ajay, you are welcome to submit your applications and ideas at [www.dadp.com](http://www.dadp.com) under the section 'Submissions'. You can also write to us at [dadp.query@relianceinfo.com](mailto:dadp.query@relianceinfo.com) with the details.

**C**an you publish a list of colleges under your University Relations Programme so that we can participate in application development for Reliance through URP?

**Anish Trivedi - Faridabad**

**Editor:** Dear Anish, you can find more details on URP from [http://www.dadp.com/dadp/html/university/university\\_relations.html](http://www.dadp.com/dadp/html/university/university_relations.html). Do write to us at [dadp.query@relianceinfo.com](mailto:dadp.query@relianceinfo.com) in case you need any more details.