

August 2006

R-EVOLUTION

Newsletter for the Developer Community

For Private Circulation only



Editorial

Dear Developers:

It was very interesting to read the responses to our 'Lead Story' in the June 2006 issue of *R-evolution* which talked about content ruling mobile data services worldwide. It depicts the high level of awareness and inclination among developers to create unique and appealing mobile applications using a range of content available across genres in India.

Content related to entertainment, music, news & information, religion and devotion, have huge audiences in our country and therefore their consumption is very high. This behaviour can be explored to create compelling services for end users. With increasing number of handsets supporting rich multimedia services and networks facilitating streaming media and other advanced services, the time is ripe for more innovations from developers and content providers to make the most of the opportunities.

The 'Lead Story' of this Month has been written by Tejal Trivedi of the Content team of the Applications and Solutions Group, Reliance Communications. Her article talks about WAP Technology, the vital link between telecommunications systems and the Web. I am sure you will find the article quite interesting and informative.

The 'Tech Tip' section illustrates the operations using the file connection APIs in J2ME.

Keep writing in with your feedback and suggestions.

Warm regards,

Saurabh Chakrabarti

Reliance Developer Programme Team

WAP WONDERLAND

Page 2

WAP PROTOCOL STACK



LEAD STORY
WAP Wonderland
Page 2



TECHTIP
File Connection
Page 4



BRAINGYM
Test your coding skills
Page 7



FEEDBACK
Letters to the Editor
Page 7

THE WAP WAY TO A MOBILE WONDERLAND

Tejal Trivedi talks about the flexibility and benefits of Wireless Application Protocol (WAP) in the world of mobile applications and technology.



LEAD STORY

The WAP Architecture



The term Wireless Application Protocol (WAP) may be jargon to a layman, but to the mobile technology provider is a tool to introduce a wonderful world of interactive applications.

WAP is a secure specification that allows users to access information instantly from handheld wireless devices. The basic function of WAP is to enable mobile phones to communicate with a server in a network. It brings both the telecommunications systems and the Web together.

WAP technology can be built on any operating system and information can be available in push-and-pull formats which allow users to interact with services via voice and data interfaces.

WAP can be used to download abstract data types—almost anything available on the Web like weather, news, stock market information, product updates, and software downloads.

There is also a broad spectrum of other existing WAP applications like e-mail, information search and retrieval, e-Commerce, calendar and office applications, maps and position-based services, voice messaging and communication, call control, and Intranet/Internet access.

Most wireless networks and mobile phones in India support WAP, which is an International standard.

The WAP system

The WAP system works when a request is made in Wireless Markup Language (WML). The request is passed through a WAP Gateway which retrieves the information from an Internet server.

The requested information is then sent from the WAP Gateway to the WAP client, using the appropriate mobile network.

This form of technology has a lot of advantages. The most important is the ability to allow multi-party communication. WAP makes it possible for several mobile phone users to communicate even while on the move using different devices. Instant and mobile voice, text, image and position messaging capabilities make this medium extremely powerful.

Applications on WAP

The use of WAP allows a user to access a number of useful applications. The mobile phone can become a personal pocket guide. Regardless of a user's location, a WAP-enabled mobile phone can act as a personal guide to the city. Users can plan their evening entertainment, reserve seats at a restaurant, request tickets for a local art show, book cinema tickets, or locate a restaurant for a quiet drink.

Some useful applications:

Yellow pages - With WAP-enabled mobile phones one can have every Yellow Pages directory in the world at one's fingertips. Users can locate and avail local services such as taxi services from anywhere because they have thousands of useful telephone numbers virtually in their pockets. This will considerably save their precious time.

Stock quotes - An automatic WAP-based stockbroker tells users the prices of shares. The user can carry out the entire transaction over the Web without moving from the spot.

Weather report - An updated world-wide detailed weather report is always available if the mobile phone is WAP-enabled. Users can be alerted on storm warnings and possible heat waves.

Other entertainment - Astrology applications can give the users constant round-the-clock access to an up-to-the-minute horoscope. Sound bytes and video clips can keep users entertained at a traffic jam or inside an elevator.



WAP in Reliance Communications

Reliance Mobile World provides Value Added Services (VAS) to Reliance Mobile customers with the use of WAP. A team called Applications Solutions and Content Group started working on developing WAP-enabled content. The first challenge for the group was to develop a new WAP interface. The technical team got an interface ready in less than a month.

After studying various other WAP sites, the good ones were benchmarked. Initially the content to be provided was text, images, and ring tones. Other content like wallpapers, video clips, true tones, and voice ringtones were soon added. The current menu includes sections like Spotlight, News, Videos, Movie Masala, Sports, Music, Wall Papers, Animation, Astrology, Spirituality, and Know your country. Over time we can expect the bandwidth required by applications to increase steadily. Therefore, there is a need to optimise WAP-enabled devices and network resources for wireless environments. In future we can also expect mobile operators to keep optimising WAP support for richer multimedia applications.

About the author



Tejal Trivedi works in the Content team of Reliance Mobile World, Reliance communications, DAKC. She likes to work on multimedia applications and her hobbies include listening to music, reading books, and surfing the Web.

USING FILE CONNECTION APIs IN J2ME



TECH TIP

I/O operations in J2ME devices are handled using the Generic Connection Framework (GCF) by means of connection interface implementations specific to each connection type. The different connection extensions are built using a URL adequate to the different connection types such as `http://`, `sockets://`, and so on.

In principle, the GCF is general enough to support connections to files but this has never been a mandatory part of J2ME or MIDP and in general it has been left out of most implementations. Even if such a kind of connection would be built, there is a lack of support for file specific operations such as

rename or delete. In addition, access to local files has important implications with respect to security, privacy, and system stability that have to be taken into account.

The FileConnection API [JSR-075] fills the gap by giving access to file systems and support for file-oriented operations. The API assumes the existence of a file system in the device that can be located. For example, in removable memory cards, flash memory, or other types of persistent storage. This API is not meant to be a replacement for the Record Management System (RMS) but rather a complement to it allowing MIDlets to interact with native applications.

System property

Since the FileConnection API is an optional extension, a system property has been added to indicate the API's presence. The `microedition.io.file.FileConnection.version` system property contains the implemented version of the API. Currently this property should have the value 1.0 to indicate the current status of the API or null if the API is not present. Another useful system property is `file.separator`, which contains the character used to separate directories, typically with the value `'/'`.

The API is very simple containing one class, two interfaces, and two exceptions. The most important part is the FileConnection interface, which extends the connection interface and gives access to directories and individual files. Implementations of

FileConnection are created using the `Connector.open()` method. The argument of the `open()` method is a URL with the format `file://<host>/<path>`.

Security implications

When developing an application using the FileConnection API, it is important to take into account the security implications of the API. File operations are restricted with the aim of protecting the user's private data and the overall system security. File operations can be executed only if the needed permission has been acquired before; otherwise a `SecurityException` will be thrown. It is important to be aware of this and include a catch `SecurityExceptions` statement when appropriate.

A simple implementation

Here is an example of a simple implementation of File Connection API. This sample code is used for accessing local file system images for Nokia Devices.

```
import java.io.*;
import java.util.Enumeration;
import javax.microedition.io.Connector;
import javax.microedition.io.file.FileConnection;
import javax.microedition.io.file.FileSystemRegistry;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

public class getimage extends MIDlet implements
CommandListener
```




```

        lstDirectory.append(s + s1, null);
        fileconnection =
(FileConnection)Connector.open(s + s1, 1);
        inputstream =
fileconnection.openInputStream();
        l = fileconnection.fileSize();
        lstDirectory.append("Ava:" + l, null);
        abyte0 = new byte[(int)l];
        inputstream.read(abyte0);
        lstDirectory.append("len:" + abyte0.length,
null);
        image = Image.createImage(abyte0, 0,
abyte0.length);
        form = new Form("Hello");
        form.append(image);
        lstDirectory.append("come", null);
        Display.getDisplay(this).setCurrent(form);
        flag = false;
        if(!flag)
            break;
        lstDirectory.append(s1, null);
    } while(true);
    fileconnection.close();
}

void changeToDirectory(String s)
{
    System.out.println("comign 1");
    if(s.equals(".."))
    {
        char c = fullPath.charAt(0);
        int i = fullPath.lastIndexOf(c, fullPath.length()
- 2);
        fullPath = fullPath.substring(0, i + 1);
    } else
    {
        System.out.println("comign 2");
        System.out.println("fullpath:" + fullPath);
        fullPath += s;
        System.out.println("fullpath:" + fullPath);
    }
    if(fullPath.length() == 1)

```

```

        getRootDir();
    else
        try
        {
            buildFileList(fullPath);
        }
        catch(Exception exception)
        {
            showError(exception.getClass() + excep-
tion.getMessage());
            return;
        }
    }

    void displayFileProperties(String s, String s1)
    {
        try
        {
            FileConnection fileconnection =
(FileConnection)Connector.open(s, 1);
            Alert alert = new Alert(s1, "Last Modified " +
new Date(fileconnection.lastModified()) + "\n" +
"Write access: " + (fileconnection.canWrite() ? "yes"
: "no") + "\n" + "Read access: " +
(fileconnection.canRead() ? "yes" : "no") + "\n" +
"File size: " + fileconnection.fileSize(), null,
AlertType.INFO);
            alert.setTimeout(-2);
            Display.getDisplay(this).setCurrent(alert);
            fileconnection.close();
        }
        catch(IOException ioexception) { }
    }

    public void commandAction(Command command,
Displayable displayable)
    {
        if(command == cmSelect)
        {
            String s =
lstDirectory.getString(lstDirectory.getSelectedIndex(
));

```

```

            if(s.endsWith("/") ||
s.equals(".."))
                changeToDirectory(s);
            else
                displayFileProperties("file://"
+ fullPath + s, s);
        } else
        if(command == cmExit)
        {
            destroyApp(false);
            notifyDestroyed();
        }
    }

    void showError(String s)
    {
        Form form = new Form("error");
        form.append(s);
        form.setCommandListener(this);
        form.addCommand(cmExit);
        Display.getDisplay(this).setCurrent(form);
    }

    private static final String ROOT = "/";
    private static final String DIRECTORY_INDICA-
TOR = "/";
    private static final String UP_DIRECTORY_INDI-
CATOR = "..";
    private String fullPath;
    List lstDirectory;
    Image imgDirectory;
    Image imgFile;
    Image imgUpDirectory;
    private Command cmExit;
    private Command cmSelect;
    private Command cmBack;
}

```

For clarifications mail dadp.newsletter@relianceada.com.



QUIZ

BRAINGYM

1. Comment lines are:

- a) Compiled in the object code
- b) Ignored by the compiler
- c) Included in the executable code
- d) Ignored by the linker

2. A syntax error is signaled by the:

- a) Compiler
- b) Linker
- c) Editor
- d) Run time system

3. When a program is running, the value of a constant:

- a) Can be changed
- b) Cannot be used
- c) Is always hidden
- d) Cannot be changed

4. The type of a function/procedure/module is determined by:

- a) Its arguments
- b) The value returned
- c) Its name
- d) The address returned

5. Which type of code is editable?

- a) Object
- b) Executable
- c) Library
- d) Source

Answers: 1. a 2. a 3. d 4. a 5. d

Monthly poser

In which research lab was the programming language C developed?

- | | |
|--------------|---------|
| 1) IEEE | 3) AT&T |
| 2) Microsoft | 4) Bell |

Answer to June's month's poser: C with classes

The winner of June's poser (through a lucky draw) is **Ganesh P**, 4, Rathina Singh Kulam Street, Arani Road, Velapadi, Vellore 632001, Tamil Nadu. Winner will receive prize by courier.

Answers to the Monthly Poser should be sent to dadp.newsletter@relianceada.com mentioning Monthly Poser—August 2006 as the subject with the sender's location. 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@relianceada.com. Please note that contributions may be edited for clarity, style or length.

Editorial Team : Saurabh Chakrabarti, Krishna Udipi, 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 Communication, DAKC, Navi Mumbai



FEEDBACK

Letters to the Editor

I liked reading the article titled, 'Content – The soul applications' in the June 2006 issue of *R-evolution*. It gives useful insight on content and its importance in shaping the mobile data services business for operators. It mentions an important perspective that mobile users will soon have access to much more interesting content like TV, music, and interactive shows.

Sumit Sen, Kolkata

Editor: Dear Sumit, thanks for the feedback and comments. We will try to bring out more such interesting and informative articles in future.

I belong to the Reliance Developer Programme community and have a few queries on XML Parsing. Who can I contact for responses?

Arpita Mongia, Patna

Editor: You are welcome to send your queries to us by e-mail at dadp.query@relianceada.com for a quick response. You can also post your queries at the discussion forum on www.dadp.com