

## **The Following information and code is compiled by Mr. Faisal Hassan (Lecturer in CCIT) for Advance Java Technologies**

### **Introduction to Java Mail API**

The Java Mail API allows the developers to add mailing functionalities to their java applications. The Java Mail API provides a platform-independent framework to develop mail and messaging applications in Java technology. Java Mail API is also protocol-independent making it more useful. You can use many different protocols to work with mails and messages.

Java mail api is designed to provide functionality such as read, compose and send from within your java programs.

The Java Mail API 1.4.1 supports JDK 1.4 or higher version. Current version of Java Mail API can be downloaded from the link given below:

<http://java.sun.com/products/javamail/downloads/index.html>

### **Installing Java Mail API**

#### **Steps to install Java Mail**

1. Unzip the javamail-1\_4\_1.zip archive.
2. Set CLASSPATH to include the "mail.jar" file obtained from the download

After installing Java Mail API, be sure that Java Beans Activation Framework is installed. Java Mail will work with JAF 1.0.2, but it is recommend to use of JAF 1.1 (currently the newest version).

#### **Download the JAF 1.1 zip from the link given below:**

<http://java.sun.com/javase/technologies/desktop/javabeans/jaf/downloads/index.html>

#### **Steps to install Java Beans Activation Framework**

1. Unzip the jaf-1\_1\_1.zip archive.
2. Set CLASSPATH to include the "activation.jar" file obtained from the download

**Note:** JAF 1.1 is included in JDK 1.6 and JAF 1.1.1 will be included in a future JDK 1.6 update release.

#### **Possible Uses**

1. Send Email from any type of java application.
2. Composing, reading and sending electronic mail.
3. Send Email from java stored procedure.Create a GUI Email client.
4. Dealing with sending and receiving attachments with Email.

5. To search for messages.

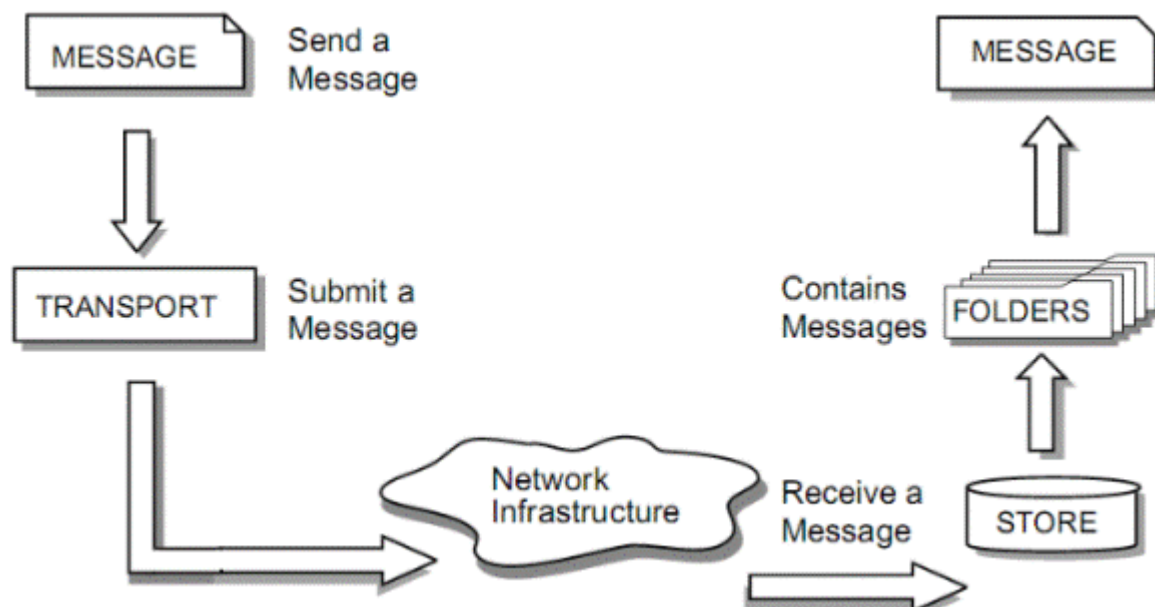
## How does Email works

### Each Internet domain has email Server, when a user sends an Email...

1. Email client Program sends the message to Email server.
2. Email server contact to the Recipient's Email server provided in the Email address.
3. Email server checks that user name is valid or not.
4. If found valid send email to the address's email server.
5. When recipient log on his mail account, gets his Email.

### Each Internet domain has email Server, when a user sends an Email...

1. Email client Program sends the message to Email server.
2. Email server contact to the Recipient's Email server provided in the Email address.
3. Email server checks that user name is valid or not.
4. If found valid send email to the address's email server.
5. When recipient log on his mail account, gets his Email.



### Mailing protocols

Basically, a protocol represents standard method used at each end of a communication channel, in order to properly transmit information.

Generally four protocols are used to send and receive Emails:

#### Simple Mail Transfer Protocol (SMTP):

SMTP is used to deliver Email to the recipient mail server. Java mail Application communicates with company's or user's ISP (Internet service provider) SMTP server, that SMTP Server will transfer the message to recipient SMTP Server.

**POP (Post Office Protocol 3):**

This protocol defines a single mailbox for a single user and a standardized way for users to access mailboxes and download messages to their computers as well. This provides facility to users to retrieve e-mail when connected. Once the messages are downloaded from the server, you can disconnect the Internet connection and read your mail.

**IMAP (Internet Message Access Protocol):**

IMAP is a client/server most advanced protocol in which Email is received and held for user by his Internet server. It supports both connected and disconnected modes of operation, allows multiple clients to access the same mailbox and provides server side searches.

**MAPI (Messaging Application Program Interface):**

MAPI is used to send Email with in windows application and provide facility to take advantage of word processors, spreadsheets, and graphics applications. One of the most popular Email applications Eudora includes a MAPI Server.

## Sending message using Java Mail

```
import java.util.*;

import javax.mail.*;

import javax.mail.internet.*;

public class mail {

    public static void main(String args[]) throws Exception {

        String host = "mail.internetspell.com";

        String from = "test@internetspell.com";

        String pass = "12345";

        Properties props = System.getProperties();

        props.put("mail.smtp.host", host);

        props.put("mail.smtp.user", from);

        props.put("mail.smtp.password", pass);

        props.put("mail.smtp.port", "25");

        props.put("mail.smtp.auth", "true");
```

```

String[] to = {"ccit@hotmail.com"}; // added this line

Session session = Session.getDefaultInstance(props, null);

MimeMessage message = new MimeMessage(session);

message.setFrom(new InternetAddress(from));

InternetAddress[] toAddress = new InternetAddress[to.length];

// To get the array of addresses

for( int i=0; i < to.length; i++ ) { // changed from a while loop

    toAddress[i] = new InternetAddress(to[i]);

}

System.out.println(Message.RecipientType.TO);

for( int i=0; i < toAddress.length; i++) { // changed from a while loop

    message.addRecipient(Message.RecipientType.TO, toAddress[i]);

}

message.setSubject("sending in a group");

message.setText("Welcome to JavaMail");

Transport transport = session.getTransport("smtp");

transport.connect(host, from, pass);

transport.sendMessage(message, message.getAllRecipients());

transport.close();

}

}

```

## Reading message using Java Mail

```
import java.io.*;

import java.util.*;

import javax.mail.*;

public class ReadMail {

    public static void main(String args[]) throws Exception {

        String host = "mail.internetspell.com";

        String user = "test@internetspell.com";

        String password = "12345";

        // Get system properties

        Properties properties = System.getProperties();

        // Get the default Session object.

        Session session = Session.getDefaultInstance(properties);

        // Get a Store object that implements the specified protocol.

        Store store = session.getStore("pop3");

        //Connect to the current host using the specified username and password.

        store.connect(host, user, password);

        //Create a Folder object corresponding to the given name.

        Folder folder = store.getFolder("inbox");

        // Open the Folder.

        folder.open(Folder.READ_ONLY);

        Message[] message = folder.getMessages();

        // Display message.

        for (int i = 0; i < message.length; i++) {

            System.out.println("----- Message " + (i + 1) + " -----");
```

```
System.out.println("SentDate : " + message[i].getSentDate());

System.out.println("From : " + message[i].getFrom()[0]);

System.out.println("Subject : " + message[i].getSubject());

System.out.print("Message : ");

InputStream stream = message[i].getInputStream();

while (stream.available() != 0) {

System.out.print((char) stream.read());

}

System.out.println();

}

}

folder.close(true);

store.close();

}

}
```