1.Email Engine 3.1 Component Specification

Changes are in blue and additions in red.

2.Design

The Email Engine component provides basic and advanced email sending functionality through Java APIs. It is not an email reader (like Elm), a graphical front end (like Outlook or Thunderbird), a mail transfer agent (like Sendmail or Postfix), or a mail server (like SMTP or MIME servers).

Programs that need to send email notifications can do so easily through the Email Engine component. A mail transfer agent must be available to route the mail to its final destination and the configuration file specifies where the transfer agent is (most ISPs provide and require the use of their own transfer agent).

Mail senders can now also mark the priority of their mail. (This is the 3.0 update)

All the original sequence diagrams had only two to three objects (as would the setPriority addition) so none are required and none have been added or reproduced.

This component must continue to meet original requirements and pass unit tests or have appropriate tests upgraded (requirement 1.1.1). Message priority headers must be added as described in the Algorithms section and the new (red) method and variable documentation (requirement 1.1.2).

Since tags and updated changelogs in the javadoc must be set in javadoc.

Custom Exceptions have been added (there were no custom exceptions in the 2.0 version).

Changes for version 3.1:

The content type may now be specified on the TCSEmailMessage. By default, the content type will be “text/plain”, just as it was with the previous version, but now the content type may be specified to something else if needed.

2.1Design Patterns

None added

2.2Industry Standards

This component uses the JavaMail API to send email. Logging is done with log4j. Email priority settings include those specified by Microsoft and the Internet Engineering Task Force.

2.3Required Algorithms

Email priority is a setting in the headers of an email message that exists to mark a message as important for the user it is being sent to. It may be useful for critical messages like password changes and critical service interruptions.

There are three email headers commonly used to indicate priority, the “Priority” header, the “X-Priority” header and the “X-MSMail-Priority” header. The values they may take on are these (note that these values should be exact, down to spacing—there is one space between the numeral and parenthesis for X-Priority):

Priority	urgent normal non-urgent
X-Priority	1 (Highest) 2 (High)

	3 (Normal) 4 (Low) 5 (Lowest)
X-MSMail-Priority	High Normal Low

The proliferation of headers is due to the different user-agents (mail readers) that recognize different indicators of priority. In this component, we will add each one of the three available headers that matches our setting so that the maximum number of users can read our intent. (The component spec specifies Microsoft Outlook which examines X-Priority and X-MSMail-Priority) The five values will correspond with these headers:

Priority Value	Priority	X-Priority	X-MSMail-Priority
PriorityLevel.HIGHEST	urgent	1 (Highest)	High
PriorityLevel.HIGH	urgent	2 (High)	High
PriorityLevel.NORMAL	normal	3 (Normal)	Normal
PriorityLevel.LOW	non-urgent	4 (Low)	Low
PriorityLevel.LOWEST	non-urgent	5 (Lowest)	Low
PriorityLevel.NONE	N/A	N/A	N/A

The scheme with bare integers suggested in the specification has been extended and improved with public final static int variables in the TCSEmailMessage class. With PRIORITY_NONE, there will be no priority headers added.

2.4Component Class Overview

EmailEngine:

Makes the internet connection and actually sends email using the send() method. In version 3.1, the send method can now handle the content type property of TCSEmailMessage.

TCSEmailMessage:

Collects and holds the information defining an email message until it is sent by EmailEngine. In version 3.1, there is a new content type property that indicates the content type for the message body. It defaults to “text/plain”.

PriorityLevel:

The priority levels available for this component.

2.5Component Exception Definitions

SendingException:

Thrown when an email address format is invalid.

AddressException:

Thrown when there is some sort of trouble sending a message through the mail transfer agent or contacting the agent.

2.6Thread Safety

As a component usually used in a server environment, this class should be used in a thread-safe manner. Aside from possible access to non-synchronized configuration manager methods on first use of the component (can be avoided by initializing the namespace before EmailEngine uses it), the EmailEngine.send() method is thread safe as written. The TCSEmailMessage class is not thread safe and should be accessed by only one thread at a time; due to its nature in representing a single message, this is not a problem.

The version 3.1 update has no effect on thread safety.

3.Environment Requirements

3.1Environment

- Java JVM 1.2, 1.3, or 1.4.

3.2TopCoder Software Components

- Configuration Manager 1.2 (any version >= 1.2)

3.3Third Party Components

- JavaMail : 1.3 : http://java.sun.com/products/javamail/
- java Activation Framework : 1.0.2 : http://java.sun.com/products/javabeans/glasgow/jaf.html (needed for JavaMail)
- log4j : 1.2.6 or greater : http://logging.apache.org/log4j/docs/
- Xerces : http://xml.apache.org/ (needed for Configuration Manager)

4.Installation and Configuration

4.1Package Name

com.topcoder.messaging.email

(one abstract class in com.topcoder.messaging)

4.2Configuration Parameters

No changes to EmailEngine configuration have been made,

In the configuration file in com/topcoder/messaging/email/EmailEngine.xml:

Parameter	***Value***	***Example***
smtp_host_addr	Host name where the SMTP server for the transfer agent resides	“smtp.isp.com”
smtp_host_port	Port number for the SMTP server (usually 25)	“25”
username	SMTP username, usually unnecessary	“”
password	SMTP password, usually unnecessary	“”

The priority headers detailed in the algorithms section may someday be changed or updated and they can be altered in the configuration file (the configuration file will usually be empty or nonexistent). These values will override the default values, so if you want to keep using the default values, copy them into the file also. The file name is “com/topcoder/messaging/email/EmailEngine.xml”:

Parameter	***Value***	***Example***

Priorities.highest	Custom priority keys and values for this priority level.	“X-Priority2:Super High”
Priorities.high	Host name where the SMTP server for the transfer agent resides	“X-Priority2:High”
Priorities.normal	Host name where the SMTP server for the transfer agent resides	“X-Priority2:Normal”
Priorities.low	Host name where the SMTP server for the transfer agent resides	“X-Priority2:Low”
Priorities.lowest	Host name where the SMTP server for the transfer agent resides	“X-Priority2:Super Low”
Priorities.none	Host name where the SMTP server for the transfer agent resides	Empty value, “”.

4.3Dependencies Configuration

An external mail transfer agent (like Sendmail or Postfix) must be set up (this is usually done by your ISP) and the configuration file must point to it.

5.Usage Notes

5.1Required steps to test the component

- Extract the component distribution.
- Follow dependencies configuration.
- Execute ‘ant test’ within the directory that the distribution was extracted to.

5.2Required steps to use the component

Set up the mail transfer agent and configuration file as described, create a TCSEmailMessage, and send it using EmailEngine.

5.3Demo

Sending A High Priority Email:

```
TCSEmailMessage message = new TCSEmailMessage();
message.setSubject(“High Priority”);
message.setBody(“Pretty Important\n-Writer\n”);
message.setFromAddress(“sender@topcoder.com”,
     “My Name”);
message.setToAddress(“recipient@client.com”,
     “Your Name”);
```

```
// This is the new method call.
message.setPriority(PriorityLevel.HIGHEST);
// For an HTML message rather than a plain text message,
// the content type should be changed.
message.setContentType(“text/html”);

EmailEngine.send(message);
```

It's as simple as that!

Here is a sample configuration file for custom email priority settings.

```
<?xml version="1.0" encoding="UTF-8"?>
<CMConfig>
  <Property name=”Priorities”>
    <Property name=”normal”>
      <!--This will remove the default headers and replace
         them with this header only for the “normal” setting
         and not for the others.  The headers set will be
         “X-Priority2:SuperNormal” and
         “Z-Priority:Normal” -->
      <Value>X-Priority2:Super-Normal</Value>
      <Value>Z-Priority:Normal</Value>
    </Property>
    <Property name=”none”>
      <!--This will set a header for none, even through there
        is no default header for none.  The header set will
        be “X-Priority:No Priority” -->
      <Value>Priority:No Priority</Value>
    </Property>
    <Property name=”high”>
      <!-- Don't use default property entries for this property
        but don't use any custom ones either.  The empty Value
        here is included as a placeholder for
        configuration Manager-->
      <Value></Value>
    </Property>
  </Property>
</CMConfig>
```

6.Future Enhancements

More specific headers could be identified for users to set.