Contactless API

Contactless Communication API (JSR 257) defines a Java ME optional package that allows developer to access contactless communication, bi-directional communication and accessing read-only information using differente technologies via short-range wireless means, such as RFID or barcodes, infrared light, or even Bluetooth, and presents such devices via a simple, homogeneous interface to Java applications.

RFID in Brief:

- RFID (Radio Frequency IDentification) is a technology to carry information over short range by radio waves
- 13.56 MHz frequency used in mobile devices
- A tag (transponder) contains digital information in a microchip
- A reader communicates with a tag
- Two types of tags
- Active tag has own power source and longer distance
- Passive tag gets power from the incoming signal

NFC and NFC Forum

- NFC (Near Field Communication) specifies simple wireless communication between close coupled devices
- NFC enables
  - Establishing other types of wireless communication between devices
  - Compatibility with existing contactless smart cards
- NFC Forum defines common protocols for basic links between NFC enabled devices* NFC Forum Data Exchange Format (NDEF) provides vendor independent structure for data

Visual Tags in Brief

- Optically machine readable information on printed material, typically in the form of bar codes or data matrices
- Widely used in product identification
- Symbology defines the features of the visual tag
- Used character set
- Encoding and decoding rules
- Data size
- Error checking
- Printing requirements
- Over 200 known symbologies, only few widely used
- UPC / EAN / JAN in article numbering

Use Cases for JSR 257

- Read a URL to movie web page from a tag
- Store personal shortcuts like phone numbers to a tag
- Bluetooth or WLAN connection initiation in a multiplayer game with RFID communication* Set device access point settings from a tag
- Field force on-the-job reporting using RFID tags* Data gathering from RFID tags to a server

Registering for Target Discovery

```
//Get DiscoveryManager instance and set TargetListener
//for NDEF_TAG target
DiscoveryManager dm = DiscoveryManager.getInstance();
try {
    dm.addTargetListener(this, TargetType.NDEF_TAG);
} catch (ContactlessException ce) {
    // handle exception
}
```

Registering for NDEF - Record Discovery

```
// Get DiscoveryManager instance and set TargetListener
// for NDEF_TAG and NFC_PEER targets
DiscoveryManager dm = DiscoveryManager.getInstance();
try {
     NDEFRecordType recordType = new NDEFRecordType(NDEFRecordType.NFC_FORUM_RTD, "MyOwnType");
     dm.addNDEFRecordListener(this, recordType);
} catch (. . .) {
     // handle exception
}
```

Making a Connection to the Target

```
public void targetDetected(TargetProperties[] prop) {
// Select first target
```


(C) Copyright Nokia 2013. All rights reserved.

```
TargetProperties target = prop[0];
try {
     // NDEF_TAG target found
     String url = target.getUrl();
     // Open NDEFTagConnection to the target
     conn = (NDEFTagConnection)Connector.open(url);
     // Read data from the target
     NDEFMessage message = conn.readNDEF();
     NDEFRecord[] records = message.getRecords();
} catch (. . .) {}

}
```

Receive Read-Only Data from NDEF Tag

```
public class MyMIDlet extends MIDlet implements NDEFRecordListener {

   public void recordDetected(NDEFMessage ndefMessage) {
      NDEFRecord[] records = ndefMessage.getRecords();
      for (int i=0; i&lt;records.length; i++) {
           // Handle data
      }
   }
}
```

Read Visual Tag Image

```
public void readVisualTag() {
   checkReadSymbologySupport();
   try {
        String[] images = SymbologyManager.getImageClasses();
        // Open connection to visual tag
        VisualTagConnection conn = (VisualTagConnection)
        Connector.open("vtag://");
        String data = conn.readVisualTag(getImage(), images[0], mySymbology);
        // Handle data from the image
        conn.close();
    } catch (. . .) {
        // handle exception
    }
}
```

Generate Visual Tag Image

```
public void generateVisualTag() {
   checkReadSymbologySupport();
   try {
       // Get properties for symbology
       ImageProperties properties = SymbologyManager.getImageProperties("code-39");
       String imageClass = getImageClass();
       VisualTagConnection conn = (VisualTagConnection)
       Connector.open("vtag://");
       // Generate visual tag image
       Object vtagImage = conn.generateVisualTag("test", imageClass, properties);
       . . .
   } catch (. . .) {
       // handle exception
   }
}
```

Summary

- JSR 257 offers communication to various contactless targets
- Can be extended to cover future contactless targets
- Provides general discovery mechanism
- Eases tasks of the application developer
- Supports MIDP push mechanism for automatic application launch for NDEF formatted data
- Flexible design and minimal set of mandatory features
- Allows implementations to support only RFID tags or visual tags or both

More Information

- JSR 257 Contactless Communication API in Java Community ProcessSM (http://jcp.org/en/jsr/detail?id=257)
- NFC Forum: http://www.nfc-forum.org/home
- Visual tag information on AIM Global (http://www.aimglobal.org/technologies/barcode/) [1]

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.