Como utilizar um RMS

O que é um RMS

Continuando a ajuda aos novatos, vou escrever um breve arquivo mostrando o caminho das pedras de como utilizar o RMS para armazenar e gerenciar informações na memória.

Para quem não sabe RMS é o acrônimo de *Record Management System*, ou em Português, Sistema de Gerenciamento de Registros. É uma classe disponível no J2ME que é muito utilizada em armazenar informações em tempo de execução do aplicativo, e que serão necessárias após o encerramento do mesmo. Ela é muito utilizada para armazenamento de informações como *High Scores* (ou Recordes), opções como som ou vibração do aplicativo.

Imports necessários

Para utilizar o rms é necessário do seguinte import

```
        import javax.microedition.rms.*;
```

Como Abrir o Record Store

O primeiro passo para a utilização de um Record Store é chamarmos o método openRecordStore para abrirmos o record

Definição:

```
openRecordStore(String recordStoreName,
 boolean createIfNecessary)
 recordStoreName = Nome do RecordStore a ser inserido]
 createIfNecessary = Cria o recordStor se o mesmo não existir
```

Exemplo:

```
RecordStore optionStore = RecordStore.openRecordStore("RecordApp",true);
```

Como receber um dado de um Record Store

Um `RecordStore` sempre trabalha com dados em formato *array de bytes*. Portanto, ao receber e ao inserir um dado é necessária a conversão do mesmo para *array de bytes*. Abaixo está um exemplo de recebimento do primeiro registro inserido em *Record Store* e sua transformação para *string* na variável `sRecord`.

Definição:

```
getRecord(int recordId)
 recordId = indice do recorde a ser deletado
```

Exemplo:

```
         ByteArrayInputStream bOptions = new
ByteArrayInputStream(optionStore.getRecord(0););
         DataInputStream dOptions = new DataInputStream(bOptions);
         sRecord = dOptions.readUTF();
```


(C) Copyright Nokia 2014. All rights reserved.

Como inserir um dado de um Record Store.

Existem 2 formas de adicionar um dado em um *Record Store*. Vale a pena lembrar que para inserir um dado no *Record Store* é necessário convertê-lo para *array de bytes*, e nos exemplos abaixo estamos transformando uma *string* em *array de bytes*.

Método setRecord: Esse método deve ser utilizado quando você deseja determinar a posição onde o dado será inserido no *Record Store*.

Definição:

```
setRecord(int recordId, byte[] newData,
 int offset,
 int numBytes)
 recordId = indice do recorde a ser inserido
 newData = array de bytes a ser inserido no recordStore
 offset = índice da primeira posição de array a ser inserido no RecordStore
 numBytes = tamanho total do array de bytes
```

Exemplo

```
optionStore.setRecord(iindex, option.toByteArray(),0,option.length);
```

Método addRecord: Esse método deve ser utilizado quando o *Record Store* irá controlar qual o próximo item vazio que será inserido o dado.

Definição:

```
addRecord(byte[] data,
 int offset,
 int numBytes)
 newData = array de bytes a ser inserido no recordStore
 offset = índice da primeira posição de array a ser inserido no RecordStore
 numBytes = tamanho total do array de bytes
```

Exemplo:

```
optionStore.addRecord(option.toByteArray(),0,option.length);
```

Como apagar um dado de um Record Store

Para apagar um dado de um *Record Store* é muito simples. Basta você acessar o método `deleteRecord()` passando o número da posição do *Record Store* a ser removido.

Definição:

```
deleteRecord(int recordId)
 recordId = indice do recorde a ser deletado
```


(C) Copyright Nokia 2014. All rights reserved.

Exemplo:

```
optionStore.deleteRecord(0);
```

Fechando o RecordStore

Após realizar algum interação com o *Record Store*, é necessário que você feche o mesmo. Essa operação deve ser realizada com maior freqüência ao inserir e apagar um dado, pois somente será confirmada a interação após fecharmos o *Record Store*.

Um erro comum ao se usar essa API é o usuário não fechar um registro após o mesmo ter sido inserido. Se ocorrer alguma saída inesperada ou acabar a bateria o dado do usuário não é salvo.

Definição

```
closeRecordStore()
```

Exemplo

```
optionStore.closeRecordStore();
```


(C) Copyright Nokia 2014. All rights reserved.