UTF

Unicode (UTF) − Stands for Unicode Translation Format. It is developed by The Unicode Consortium. if you want to create documents that use characters from multiple character sets, you will be able to do so using the single Unicode character encodings. It provides 3 types of encodings.

* **UTF-8** − It comes in 8-bit units (bytes), a character in UTF8 can be from 1 to 4 bytes long, making UTF8 variable width.
* **UTF-16-8** − It comes in 16-bit units (shorts), it can be 1 or 2 shorts long, making UTF16 variable width.
* **UTF-32** − It comes in 32-bit units (longs). It is a fixed-width format and is always 1 "long" in length.

The writeUTF() method of the java.io.DataOutputStream class accepts a String value as a parameter and writes it in using modified UTF-8 encoding, to the current output stream. Therefore to write UTF-8 data to a file −

The readUTF() method of the java.io.DataOutputStream reads data that is in modified UTF-8 encoding, into a String and returns it. Therefore to read UTF-8 data to a file –

import java.io.DataInputStream;

import java.io.EOFException;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.DataOutputStream;

import java.io.FileOutputStream;

public class Read\_Write\_UTF {

public static void main(String args[]) {

FileOutputStream fileOut = null;

DataOutputStream outputStream = null;

FileInputStream fileIn = null;

DataInputStream inputStream = null;

StringBuffer buffer = new StringBuffer();

try {

//Instantiating the FileOutputStream class

fileOut = new FileOutputStream("D:\\utfText.txt");

//Instantiating the DataOutputStream class

outputStream = new DataOutputStream(fileOut);

//Writing UTF data to the output stream

outputStream.writeUTF("నా పేరు నితిన్ నగర్| మీరు అతని పేరు ఏమిటి? ");

outputStream.flush();

System.out.println("Data inserted into the file");

//Instantiating the FileInputStream class

fileIn = new FileInputStream("D:\\utfText.txt");

//Instantiating the DataInputStream class

inputStream = new DataInputStream(fileIn);

//Reading UTF data from the DataInputStream

while(inputStream.available()>0) {

buffer.append(inputStream.readUTF());

}

}

catch(EOFException ex) {

System.out.println(ex.toString());

}

catch(IOException ex) {

System.out.println(ex.toString());

}

System.out.println("Contents of the file: "+buffer.toString());

}

}

Flush () method:

The **flush ()** method of OutputStream class is used to flush the content of the buffer to the output stream. A buffer is a portion in memory that is used to store a stream of data(characters). That data sometimes will only get sent to an output device, when the buffer is full.

The flush method of OutputStream does nothing.