



Java 2 Micro Edition (J2ME)



- Til apparater med begrænset hukommelse.
- Har en lagdelt struktur:

| |
|---|
| Profil (klasser rettet mod apparatet) |
| Konfiguration (virtuel maskine og basisklasser) |
| Styresystem |

- Kan derfor tilrettes mod en række apparater
- Mest udbredt er den til mobiltelefoner
 - MIDP (Mobile Information Device Profile)
 - CLDC (Connected Limited Device Configuration)
 - Delmængde af klasser fra java.lang, java.io og java.util
 - Meget kendt fra annoncer om 'fede JAVA-spil'
- J2ME er platformsuafhængigt (per apparattype)



Midletter – mobiltelefon'apletter'



Eksempel: En midlet, der viser en (fiktiv) vejrudsigt:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class Vejrmidlet extends MIDlet
{
    // systemet starter midletten
    public void startApp()
    {
        // opret et skærbillede (en liste)
        List sb = new List("Vejret", List.IMPLICIT);
        sb.append("Det bliver let skyet og blæsende", null);
        sb.append("Temperatur mellem 17 og 22 grader", null);

        // vis skærbilledet
        Display.getDisplay(this).setCurrent( sb );
    }

    // systemet standser midletten
    public void pauseApp() {}

    // systemet smider midletten væk
    public void destroyApp(boolean unconditional) {}
}
```



Klassen Display repræsenterer den fysiske skærm.

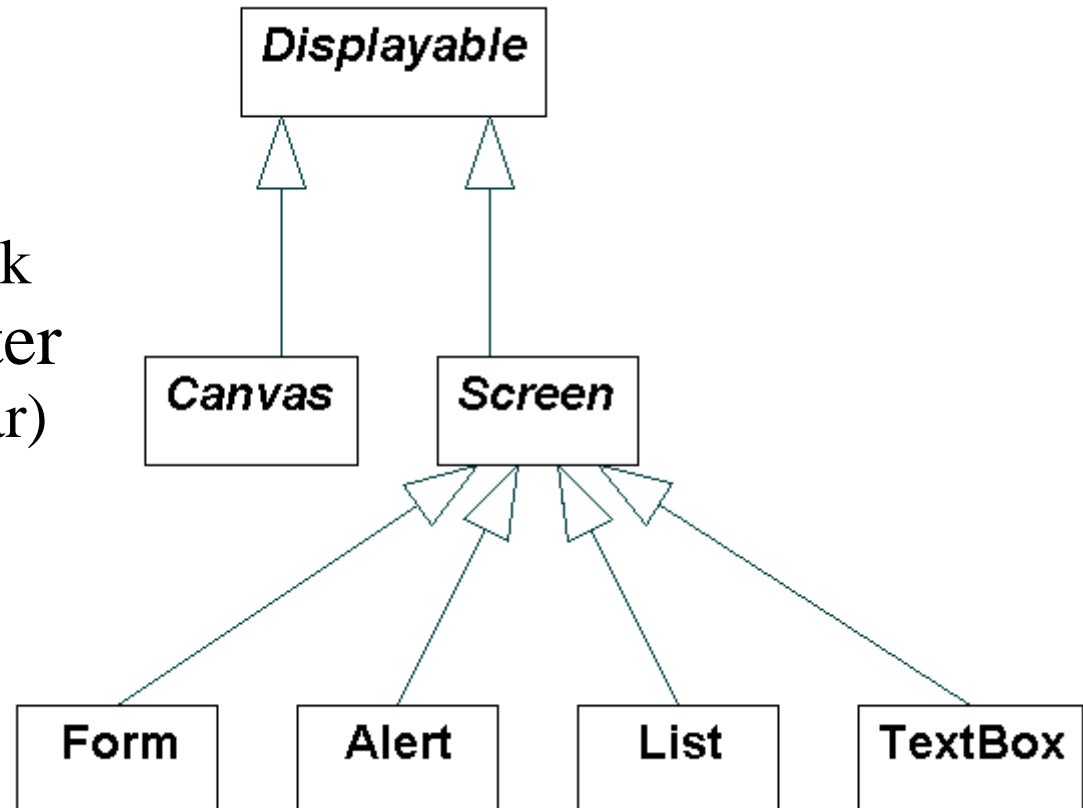
| | |
|------------------------------------|--------------------------------|
| static Display getDisplay(midlet) | skaffer Display-objektet |
| int numColors() | giver antallet af farver |
| void setCurrent(Displayable skærm) | viser et bestemt skærbillede |
| Displayable getCurrent() | giver det aktuelle skærbillede |
| boolean vibrate(int varighed) | vibrerer med telefonen |

Midletten kan skifte mellem forskellige skærbilleder ved at kalde `setCurrent(Displayable)` på Display-objektet.



Brugergrenseflader i midletter

- Det centrale element: Et skærbillede
 - Klasse, der arver fra Displayable
 - Kun ét skærbillede er synligt ad gangen.
- To måder at programmere på:
 - Med paint()-metode
 - Nedarve fra Canvas
 - MIDP 2.0: GameCanvas
 - Det mest almindelige
 - Kan blive platformsspecifik
 - Med standardkomponenter
 - Form (indtastningsformular)
 - Alert (til 'popop')
 - List (liste – evt med valg)
 - TextBox (redigere tekst)
 - Begrænsede muligheder
 - Platformusafhængig





Canvas - direkte grafik og spil



```
import javax.microedition.lcdui.*;

public class Canvasgrafik extends Canvas
{
    private int x,y;

    public Canvasgrafik()
    {
        x = getWidth()/2;
        y = getHeight()/2;
    }

    public void paint(Graphics g)
    {
        // slet baggrunden
        g.setColor( 0x00ffffff ); // hvid
        g.fillRect(0, 0, getWidth(), getHeight());

        g.setColor( 0x00000000 ); // sort
        g.drawString("Brug piletasterne",0,0,
            Graphics.TOP|Graphics.LEFT);
        g.fillRect(x,y,3,3);
    }

    protected void keyPressed(int tastkode)
    {
        switch (getGameAction(tastkode)) {
            case UP:    y--; break;
            case DOWN: y++; break;
            case LEFT:  x--; break;
            case RIGHT: x++; break;
        }
        repaint(); // bed systemet kalde paint
    }
}
```



```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class VisCanvasgrafik extends MIDlet
{
    Canvasgrafik grafik = new Canvasgrafik();

    public void startApp()
    {
        Display.getDisplay(this).setCurrent(grafik);
    }

    public void pauseApp() {}
    public void destroyApp(boolean unconditional) {}
}
```



Udviklingsværktøjer

- J2ME Wireless Toolkit fra Sun (*anbefales*)
 - Nemt at installere
 - Programmering kan gøres i JBuilder (ved at skifte JDK)
 - Kørsel sker fra Wireless Toolkit
- JBuilder Mobile Edition (*kun til meget interesserede*)
 - JBuilder Personal skal afinstalleres
 - Svært at installere
 - GUI-designer

The screenshot displays two software windows. The foreground window is JBuilder 7, showing a project named 'TestMidlet.jpx' with files 'Displayable1.java' and 'MIDlet1.java'. The GUI designer shows a clock displaying '15:38' and a gauge. The background window is the J2ME Wireless Toolkit, showing a 'New Project' dialog with 'Device: DefaultColorPhone'.

| name | value |
|----------|--------|
| name | gauge1 |
| label | |
| maxValue | 10 |
| value | 0 |



Udvikle med Wireless Toolkit

- Installér og prøv Wireless Toolkit
 - Hent fra <http://java.sun.com/products/j2mewtoolkit>
 - MIDP 1.0.4 kører i nuværende telefoner (f.eks. Nokia 3510i)
 - MIDP 2.0 er til de fremtidige telefoner
 - Kør 'ktoolbar' og vælg 'Open Project...' og f.eks. 'games'
 - Vælg 'Run' og f.eks. 'WormGame'
- Start JBuilder Personal og opret tilsvarende projekt
 - Sørg for at stien til kildetekst (source path) peger på kildeteksten!
 - F.eks. 'apps/games/src'
 - Vælg nyt JDK og vælg sti til roden af Wireless Toolkit
 - Udvid 'example.wormgame' under projektet og åbn WormPit.java
 - Ret i paint(), tilføj f.eks. `g.drawString("Hej Jacob",0, 0, g.TOP|g.LEFT)` - og gem!
- Afprøv din ændring Wireless Toolkit
 - Vælg 'Build' og 'Run'
- Opgave: Kurvetegning på en telefon
 - Eller find selv på en opgave!



Menyer og kommandoer



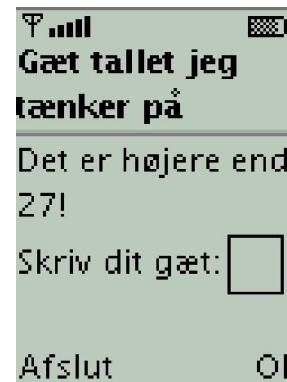
- Et Command-objekt for hver mulig handling

- Kommandoer har:

- navn (vist på skærmen)
- længere tekst (valgfri)
- type
- prioritet (m.h.p. placering)

- Tilknyttes skærbilledet

```
skærbillede.addCommand( okCommand );
```



- Menu med kommandoer vises i bunden af skærmen

- Hvordan afhænger af telefonen

- Lytter (skal implementere CommandListener)

```
skærbillede.setCommandListener( lytter );
```

- Lytter bliver kaldt når bruger vælger handlingen

```
public void commandAction(Command c, Displayable s) {  
    if (c == afslutCommand) {  
        destroyApp(false);  
        ...  
    }  
}
```

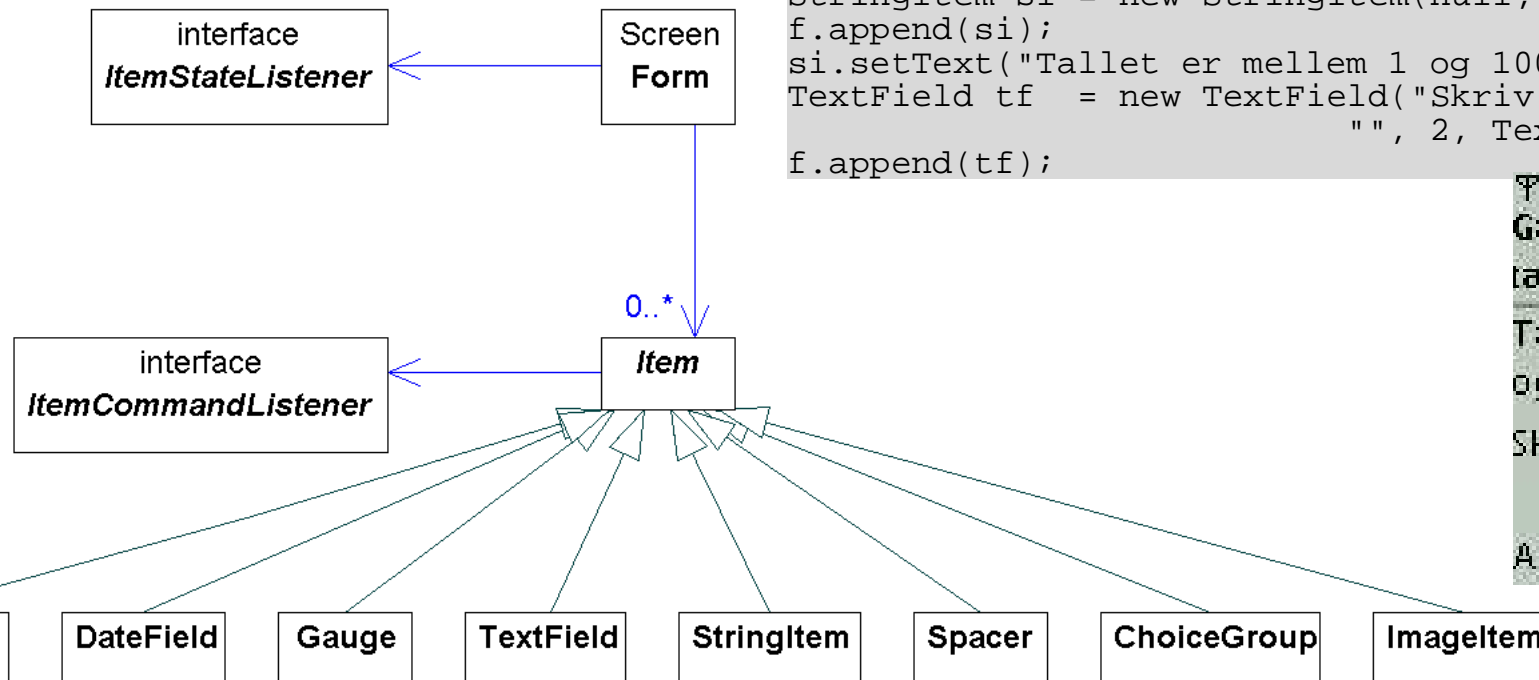


Grafiske komponenter i midletter



- Form er din ven :-)
 - Indeholder 'items' (indgange)
- To måder at opfange brugerens handlinger på
 - Lytte på formularen og få ændrede indgange at vide
 - Sætte kommandoer på formular/de enkelte indgange

```
Form f = new Form("Gæt tallet jeg tænker på");
StringItem si = new StringItem(null, null);
f.append(si);
si.setText("Tallet er mellem 1 og 100.");
TextField tf = new TextField("Skriv dit gæt:",
                             "", 2, TextField.NUMERIC);
f.append(tf);
```





Eksempel - gæt et tal



Afsnit 13.4.1 på
<http://javabog.dk/VP>

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class GaetEtTalMidletMIDlet extends MIDlet implements CommandListener {
    private Display display = Display.getDisplay(this);

    private Form f          = new Form("Gæt tallet jeg tænker på");
    private StringItem si   = new StringItem(null, null);
    private TextField tf    = new TextField(null, "", 2, TextField.NUMERIC);

    private Command afslutCommand = new Command("Afslut", Command.SCREEN, 2);
    private Command okCommand     = new Command("OK", Command.SCREEN, 1);
    private Command nytSpilCommand = new Command("Nyt spil", Command.SCREEN, 1);

    private Random tilf = new Random();
    private int tallet;
    private int forsøg;

    public GaetEtTalMidletMIDlet() {
        f.addCommand(afslutCommand);
        f.addCommand(okCommand);

        f.setCommandListener(this);

        f.append(si); // tilføj strengen til formularen
        f.append(tf); // tilføj indtastningsfeltet
    }

    private void nytSpil() {
        tallet = Math.abs(tilf.nextInt()) % 100 + 1;
        forsøg = 0;
        si.setText("Tallet er mellem 1 og 100.");
        tf.setLabel("Skriv dit gæt:");
    }

    public void startApp() {
        nytSpil();
        display.setCurrent(f);
    }

    public void pauseApp() { }

    public void destroyApp(boolean unconditional) { }

    /**
     * Reager på kommandoerne ok, afslut og nyt spil.
     * Metode skal defineres fordi klassen implementerer CommandListener-interf.
     * og den bliver kaldt fordi formularen har fået tilføjet klassen som lytter.
     * Ved afslut ryddes op og midletten giver besked på at den skal smides væk.
     */
    public void commandAction(Command c, Displayable s) {
        if (c == afslutCommand) {
            destroyApp(false);
            notifyDestroyed();
        }
        else if (c == okCommand) {
            int gæt = Integer.parseInt( tf.getString() );
            tf.setString("");
            forsøg++;
            if (gæt < tallet) si.setText("Det er højere end "+gæt+"!");
            else if (gæt > tallet) si.setText("Det er lavere end "+gæt+"!");
            else {
                si.setText("Rigtig på "+forsøg+" forsøg!");
                f.delete(1); // Fjern indtastningsfeltet fra formularen
                f.removeCommand(okCommand);
                f.addCommand(nytSpilCommand);
            }
        }
        else if (c == nytSpilCommand) {
            f.removeCommand(nytSpilCommand);
            f.addCommand(okCommand);
            f.append(tf); // Tilføj indtastningsfeltet til formularen igen
            nytSpil();
        }
    }
}
```



Eksempel: Dansk-tysk ordliste



```
public class Ordcanvas extends Canvas implements CommandListener {

    public Ordcanvas() {
        setCommandListener(this);
        addCommand(KomAfslut);
        addCommand(KomHjælp);
        addCommand(KomOk);
        info.setTimeout(Alert.FOREVER);
    }

    Ordliste ord = new Ordliste();
    Alert info = new Alert("");

    Command KomHjælp = new Command("Hjælp", Command.HELP, 4);
    Command KomSlet = new Command("Slet", Command.BACK, 3);
    Command KomAfslut = new Command("Afslut", Command.ITEM, 2);
    Command KomOk = new Command("Vis", Command.EXIT, 1);

    public void commandAction(Command command, Displayable displayable) {
        System.out.println("commandAction("+command);
        if (command == KomSlet) {
            sletBogstav();
            repaint();
        } else if (command == KomHjælp) {
            info.setTitle("Hjælp");
            info.setString("Indtast et søgeord.\n2=ABC, 3=DEF, 4=GHI etc.\n");
            Display.getDisplay(Ordmidlet.instans).setCurrent(info);
        } else if (command == KomAfslut) {
            Ordmidlet.instans.afslut();
        } else if (command == KomOk) {
            info.setTitle(ord.nøgle(indeks));
            //infoTicker.setString(ord.liste[indeks][0]);
            //info.setString(ord.liste[indeks][1]);
            info.setString(ord.værdi(indeks)+"\n(dansk: "+ord.nøgle(indeks)+")");
            Display.getDisplay(Ordmidlet.instans).setCurrent(info);
        }
    }
}
```





Eksempel: Dansk-tysk ordliste



```
protected void paint(Graphics g) {
    int skærbredde = getWidth();
    int skærmhøjde = getHeight();

    g.setColor( 0xffffffff ); // hvid
    g.fillRect(0, 0, skærbredde, skærmhøjde);

    g.setColor( 0x000000 ); // sort
    Font font = g.getFont();
    int skrifthøjde = font.getHeight();
    int y = 0;
    int i = indeks;
    do {
        String nøgle = ord.nøgle(i);
        String værdi = ord.værdi(i);
        if (i==indeks) {
            int strengbr = font.substringWidth(nøgle, 0, søgetal.length());
            g.drawLine(0,y+skrifthøjde,strengbr,y+skrifthøjde);
            g.setColor(0x000080); // blå
        } else {
            g.setColor(0x000000); // sort
        }
        g.drawString(nøgle, 0, y, Graphics.TOP | Graphics.LEFT);

        int br0 = font.stringWidth(nøgle)+3;
        int br1 = font.stringWidth(værdi);

        // er der plads til at skrive noget af oversættelsen?
        if (br0 < skærbredde) {
            int x1;
            // er der plads til at skrive hele oversættelsen?
            if (br0+br1 < skærbredde) x1 = skærbredde - br1; // Så højrestil
            else x1 = br0; // ellers klip sidste del af ordet

            g.setColor(0x7F7F7F); // lys grå
            g.drawString(værdi, x1, y+1, Graphics.TOP | Graphics.LEFT);
        }
        y += skrifthøjde;
        i ++;
    } while (y<skærmhøjde && i<ord.maxIndeks());
```

```
private StringBuffer søgetal = new StringBuffer(10);
int indeks = 0;

void sletBogstav() {
    søgetal.delete( søgetal.length()-1, 1000);
    if (søgetal.length()==0) {
        removeCommand(KomSlet);
        addCommand(KomAfslut);
        addCommand(KomHjælp);
    }
    fremsøgMuligeOrd();
}

protected void keyPressed(int tast) {
    if ('1' <= tast && tast <= '9') {
        søgetal.append((char) tast);
        if (søgetal.length()==1) {
            removeCommand(KomAfslut);
            removeCommand(KomHjælp);
            addCommand(KomSlet);
        }
        fremsøgMuligeOrd();
        repaint();
    } else {
        int handl = getGameAction(tast);
        if (søgetal.length()>0) {
            removeCommand(KomSlet);
            addCommand(KomAfslut);
            addCommand(KomHjælp);
            søgetal.setLength(0);
        }
        if (handl == UP && indeks > 0)
            indeks--;
        if (handl == DOWN && indeks < ord.maxIndeks() - 1)
            indeks++;
    }
    repaint();
}
```



Netværkskommunikation



- **Http(s)Connection** - til HTTP-kommunikation
 - f.eks.: `Connector.open("http://javabog.dk")`
- **SocketConnection** - til at åbne en TCP-forbindelse
 - f.eks.: `Connector.open("socket://host.com:79")`
 - `serversocket: Connector.open("socket://:79")`
- **SecureConnection** - til Secure Socket Layer (SSL)
 - f.eks.: `Connector.open("ssl://maskinnavn.dk:79")`
- **DatagramConnection**
 - klient: `Connector.open("datagram://123.456.789.12:1234")`
 - server: `Connector.open("datagram://:1234")`
- **CommConnection** - til eventuelle serielle porte
 - f.eks.: `Connector.open("comm:1")`

```
import java.io.*;
import javax.microedition.io.*;
...
URLConnection http = (URLConnection) Connector.open(URL, Connector.READ_WRITE);
DataOutputStream out = http.openDataOutputStream();
out.writeChars("En tekst");
```



Gemme data i telefonen



- Record Management System (RMS)

Her er et eksempel på brug:

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

// Åbn en 'fil' i telefonen. Opret den hvis den ikke allerede findes.
RecordStore database = RecordStore.openRecordStore("minFil", true);

// Skaf data i form af et array af byte
String strengDerSkalGemmes = "Hej Verden";
byte[] data = strengDerSkalGemmes.getBytes();

// gem data
database.addRecord( data, 0, data.length );

// luk 'filen'
database.closeRecordStore();
```



Midletter OTA (Over-The-Air)



- JAR-fil
 - Java ARkiv = zip-fil
 - indeholder klasserne

```
unzip -l da-ty.jar
Archive:  da-ty.jar
  Length      Date    Time    Name
-----
         222  10-06-03  03:15  META-INF/MANIFEST.MF
        1434  10-06-03  03:15  ordliste/Ordmidlet.class
        5805  10-06-03  03:15  ordliste/Ordcanvas.class
        3749  10-06-03  03:15  ordliste/Ordliste.class
```

- JAD-fil
 - Tekslig beskrivelse af midlet
 - Henviser til JAR-filen
 - Skal være korrekt

```
MIDlet-Name: Dansk-tysk ordliste
MIDlet-Version: 0.0.6
MIDlet-Vendor: Nordfalken
MicroEdition-Profile: MIDP-1.0
MicroEdition-Configuration: CLDC-1.0
MIDlet-Jar-URL: da-ty.jar
MIDlet-Jar-Size: 27697
MIDlet-1: Ordmidlet, , ordliste.Ordmidlet
```

- WML-fil (WAP)
 - HTML-lignende
 - Henviser til JAD-fil

```
<wml>
<card id="kort1">
<p>
  Velkommen!<br />
  Hent en <a href="da-ty.jad" title="Da-ty">
  Dansk-tysk ordliste (JAVA)</a><br/>
<br />
</p>
</card>
<wml>
```

Åbn nu på jeres telefoner:
<http://javabog.dk/VP/index.wml>
 og hent ordliste-prg på campusnet



Wireless Application Protocol (WAP)



Hvordan det virker inde bagved

Når en telefon skal hente et dokument via WAP sker der fire ting:

1. Først kontakter telefonen en WAP-gateway og beder den hente dokumentet. Typisk er WAP-gatewayen kodet ind i telefonen af operatøren.
2. Gatewayen kontakter så webstedet, via den almindelige HTTP-protokol. Webstedet giver dokumentet tilbage. Dokumentet er typisk et WML-dokument (Wireless Markup Language - minder om HTML).
3. Dette WML-dokument parses af gatewayen (der derpå henter evt. tilhørende billeder og andet nævnt i WML-dokumentet hos webserveren)
4. WAP-gatewayen sender en komprimeret udgave af WML-dokumentet tilbage til telefonen.

Opsætning af webserveren til WML

Klares i Apache ved at indsætte følgende MIME-typer i filen httpd.conf:

```
# Til WML-sider, til billeder (i kun i formatet WBMP) og WML-script
#
# Filendelsen "wml" skal svare til MIME-type "text/vnd.wap.wml".
AddType text/vnd.wap.wml wml
AddType image/vnd.wap.wbmp wbmp
AddType text/vnd.wap.wmlscript wmls

# Til oversat WML - dette gør en WAP-gateway overflødig
AddType application/vnd.wap.wmlc wmlc
AddType application/vnd.wap.wmlscriptc wmlsc

#Til Midletter:
AddType text/vnd.sun.j2me.app-descriptor jad
AddType application/java-archive .jar
```

Bruger man Microsoft IIS må man klikke sig igennem og tilføje hver MIME-type manuelt.



WML - Wireless Markup Language



- Minder om HTML
 - Syntaks mere striks
 - Som i XHTML skal alle koder lukkes igen
 - Starter med `<wml>` og `<card>` (i st.f. `<html>` og `<body>`)
 - Et WML-dokument kan indeholde flere skærbilleder
 - Opdelt i 'kort' - som i et spil kort

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//PHONE.COM//DTD WML 1.1//EN"
    "http://www.phone.com/dtd/wml11.dtd" >
<wml>
  <card id="kort1">
    <p>
      Velkommen!<br />
      <a href="da-ty.jad" title="Da-ty">Dansk-tysk ordliste (JAVA)</a><br/>
      <br />
      Andet:<br />
      <a href="http://wap.telia.dk" title="Telia">Wap Telia</a><br/>
      <a href="http://wap.krak.dk" >Wap Krak</a><br/>
      <a href="http://wap.orange.dk" title="Her">Orange</a><br/>
      <a href="http://wap.ht.dk" title="Bus">HT</a><br/>
      <a href="#kort2" >Videre...</a>
    </p>
  </card>
  <card id="kort2">
    <p>Dette er det andet skærbillede i WML-dokumentet
      . . . .
    </p>
  </card>
</wml>
```