

Het Java 2 Micro Edition platform (J2ME) werd een paar jaar geleden met relatief weinig fanfare geïntroduceerd. Na een paar jaar groeien in de luwte lijkt het erop dat de tijd rijp is voor de doorbraak van dit kleine broertje uit de Java familie. De fabrikanten van mobiele telefoons bieden tegenwoordig standaard ondersteuning voor Java applicaties op hun al nieuwe modellen. Mobiele apparaten worden ook steeds krachtiger. Hierdoor groeien de mogelijkheden voor applicatie-ontwikkelaars en content providers. Dit artikel geeft een hernieuwde introductie op J2ME en legt daarbij de nadruk op het Mobile Information Device Profile (MIDP).



Het kleine broertje wordt groot

Hernieuwde kennismaking met J2ME en MIDP

Platformonafhankelijkheid, het bekende “Write Once, Run Anywhere”, stond aan de basis van het succes van Java als ontwikkelplatform voor desktop en server software. En in de wereld van mobiele apparaten, waar de verscheidenheid aan hardware- en besturingssystemen nog veel groter is, is het wederom een doorslaggevende factor.

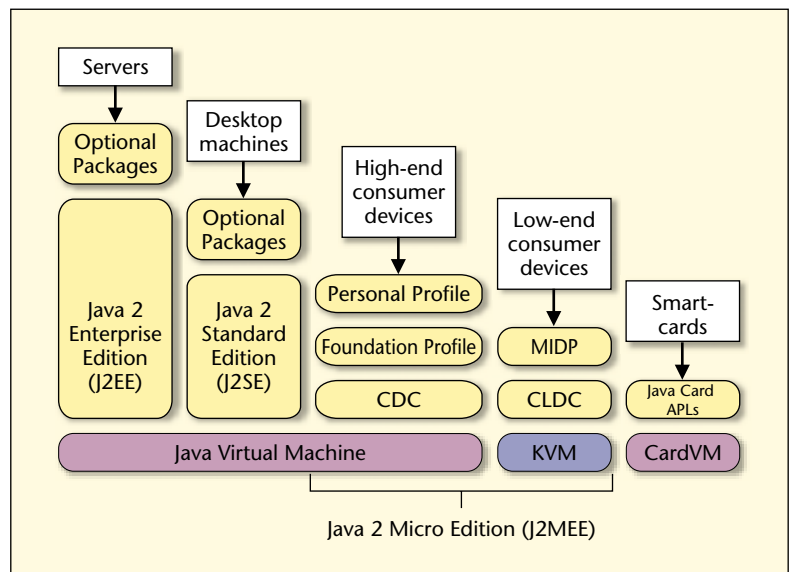
De aanblijvende vraag van de consument naar nieuwere en betere mobiele apparaten is sterk afhankelijk van het aanbod van aantrekkelijke applicaties op het gebied van communicatie, entertainment, en business. De fabrikanten van deze apparaten beseffen maar al te goed dat een populair en breed toepasbaar ontwikkelplatform onmisbaar is voor het produceren van dit soort applicaties. Het is dan ook geen verrassing dat de verschillende fabrikanten van mobiele hardware en software achter de schermen flink samenwerken op het gebied van standaardisering. Dit samenwerkingsverband heeft er onder andere toe geleid dat bijna honderd procent van alle nieuwe mobiele telefoons ondersteuning biedt voor J2ME.

ACHTERGROND Het Java platform is tegenwoordig onderverdeeld in drie edities. Elke editie is een verzameling van library's en tools gericht op een specifiek type applicatie en een specifiek type deployment-omgeving. De drie edities zijn:

- *De Standard Edition (J2SE)* richt zich op desktop omgevingen en het ontwikkelen van applicaties met

grafische user interfaces.

- *De Enterprise Edition (J2EE)* richt zich op het ontwikkelen van componenten en applicaties voor applicatie servers.
- *De Micro Edition (J2ME)* is de jongste en minst bekende telg van de Java familie. J2ME is bedoeld voor het ontwikkelen van applicaties voor kleinere apparaten met beperkte hardware mogelijkheden.



AFBEELDING 1. Een overzicht van de verschillende Java edities en de beoogde doelgebieden.

TYPES APPARATEN Het ontwerp van het J2ME platform is gebaseerd op een tweedeling van apparaat-typen op basis van hardware- en netwerkcapaciteit. Deze types zijn:

- *Mobiele apparaten met een netwerkaansluiting.*
Dit type apparaten is draagbaar en kan daardoor niet al te groot of zwaar zijn. De stroomvoorziening is meestal gebaseerd op batterijen. Dit betekent dat er minder ruimte is voor processor en geheugen en dat eventuele user interfaces relatief klein zijn. Netwerkverbindingen zijn vrijwel altijd draadloos en zowel de capaciteit als de stabiliteit van het netwerk zijn niet om over naar huis te schrijven. Mobiele telefoons, pagers, en PDA's zijn voorbeelden van apparaten in deze categorie.
- *Niet- of nauwelijks-mobiele apparaten met een netwerkaansluiting.*
Dit zijn apparaten die veelal op een vaste plek staan en die hun stroom via het stopcontact krijgen. Ze zijn groot genoeg om ruimte te hebben voor relatief krachtige processoren en een flinke hoeveelheid intern geheugen. De netwerkverbinding is meestal

- Een implementatie van een bepaalde subset van de J2SE API. Hoe beperkter de hardware, hoe kleiner de set van ondersteunde classes en packages

Voor elk apparaattype definieert J2ME een configuratie type:

- *CLDC: Connected, Limited Device Configuration voor mobiele apparaten met een netwerkaansluiting.* Deze configuratie bevat een absoluut minimale subset van de standaard classes en packages, aangevuld met een paar uitbreidingen op het gebied van netwerkcommunicatie. De CLDC configuratie draait meestal op een speciaal hiervoor ontwikkelde minimale implementatie van de Java Virtual Machine (JVM). Tot voor kort was dat meestal de zogenaamde kVM. De 'k' staat voor 'kilo', een verwijzing naar het minimale geheugengebruik. Recentelijk is deze echter vervangen door een speciaal ontworpen implementatie van de standaard "Hotspot" virtual machine. De nieuwe implementatie is stukken sneller dan de oude kVM en gaat veel efficiënter met zijn geheugen om.
- *CDC: Connected Device Configuration voor niet- of nauwelijks-mobiele apparaten met een netwerk aansluiting.* Deze configuratie draait meestal op de standaard JVM die ook op PC's wordt gebruikt en het aantal ondersteunde classes en packages is een stuk groter dan in CLDC. CDC is gelijk aan de basisfunctionaliteit van J2SE, dus zonder ondersteuning voor het bouwen van complexe user interfaces (AWT en Swing), het praten met databases (JDBC), transacties, XML, et cetera.

Een J2ME-profiel beschrijft en standaardiseert de minimale functionaliteit van aan een familie gerelateerde apparaten

breedband en stabiel. Voorbeelden van dit type apparaat zijn beeldtelefoons, draadloze media apparatuur (MP3 jukebox) en decoders voor digitale televisie.

Deze twee categorieën zullen in de toekomst steeds dichter naar elkaar toe groeien, maar voorlopig is het onderscheid nog groot genoeg om invloed te hebben op het ontwerp van een software platform als J2ME.

CONFIGURATIES In J2ME wordt de algemene hardware- en netwerkcapaciteit van een type apparaat beschreven met behulp van een *configuratie*. Een configuratie beschrijft onder andere een minimum set vereisten waaraan de hardware en de netwerkverbinding moeten voldoen. Daarnaast bevat een configuratie een specificatie van de minimaal beschikbare Java classes en packages waarmee geprogrammeerd kan worden. Een configuratie bestaat, ongeacht de hardware, uit drie onderdelen:

- Een Java virtuele machine (JVM) die de Java bytecode uitvoert
- Native code die communiceert met het onderliggende systeem

PROFIELEN Binnen de zeer brede definities van apparaattypes die worden gehanteerd door configuraties is er nog volop ruimte voor verdere onderverdeling. Een J2ME profiel beschrijft en standaardiseert de minimale functionaliteit van een bepaalde familie gerelateerde apparaten. Het doel van een J2ME profiel is om ervoor te zorgen dat alle apparaten binnen zo'n apparaatfamilie qua ondersteuning voor Java onderling inwisselbaar zijn. Software geschreven voor een bepaald profiel kan dan op alle apparaten binnen die familie (bijv. Alle mobiele telefoons) geïnstalleerd worden, ongeacht het merk of de specifieke hardware. Een profiel bevat veelal classes en packages die veel domeinspecifieker zijn dan in een configuratie. Een goed voorbeeld is ondersteuning voor user interfaces. Een mobiele telefoon heeft hele andere mogelijkheden op dat gebied dan een navigatiesysteem in een auto. Het is mogelijk voor een apparaat om meerdere profielen te ondersteunen maar een applicatie moet worden geschreven op basis van één profiel. Er bestaan profielen voor zowel CLDC als voor CDC:

- *MIDP*: Het Mobile Information Device Profile is het enige profiel voor de CLDC en is speciaal ontwikkeld met het oog op mobiele telefoons en vergelijkbare apparaten.
- *IMP*: Het Information Module Profile is een sub-set van MIDP die speciaal bedoeld is voor apparaten zonder user interface, zoals parkeermeters, alarmsystemen, en sigaretautomaten.
- *Foundation Profile*: Het foundation profiel is het basisprofiel voor CDC-gebaseerde apparaten zonder user interface.
- *Personal Basis, Personal, en RMI Profiles*: Deze profielen voegen extra functionaliteit toe aan het foundation profiel.

J2ME configuraties en profielen worden ontworpen, ontwikkeld, en beheerd door het Java Community Process. De rest van dit artikel zal verder ingaan op het schrijven van applicaties voor het MIDP profiel.

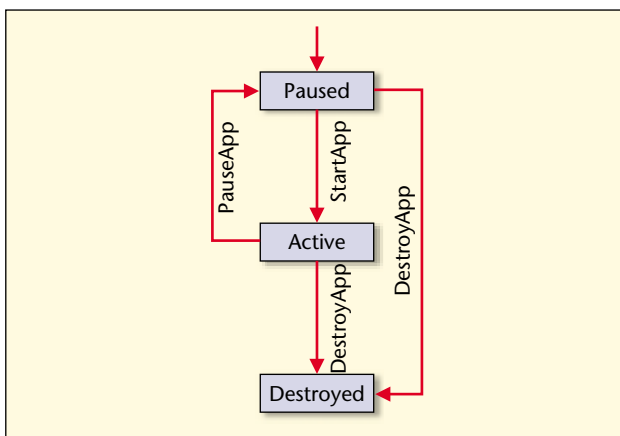
MIDP EN MIDLETS Mobiele telefoons, pagers, en PDA's hebben allemaal ongeveer dezelfde beperkingen als het gaat om processor, geheugen, en netwerk connecties. Daarnaast hebben ze ook een aantal karakteristieke beperkingen op het gebied van de grootte van het beeldscherm en het toetsenbord. Voor deze apparaatfamilie is het MIDP profiel ontwikkeld. Dit profiel biedt API's ter ondersteuning van de onder anderen de volgende activiteiten:

- Het creëren van grafische user interfaces voor kleine schermen met beperkte mogelijkheden voor het verzamelen van user input.
- Het lokaal opslaan van data.
- Het gebruiken van netwerkconnecties, met in elk geval ondersteuning voor HTTP en HTTPS.
- Het creëren van geavanceerde spelletjes

MIDP is één van de oudste J2ME profielen. De meeste moderne mobiele telefoons en PDA's ondersteunen MIDP 1.0, terwijl de allernieuwste telefoons ook al ondersteuning bieden voor MIDP 2.0.

Dit zijn de minimale eisen waaraan een MIDP 2.0 apparaat moet voldoen:

- *Scherm*
 - Schermgrootte van minstens 96 x 54 pixels
 - Ten minste 1-bit kleurdiepte (zwart-wit)
 - Schermverhouding van ongeveer 1:1
- *Input*
 - Eénhandig toetsenbord (telefoon keypad), of
 - Twee-handig keyboard, (QWERTY toetsenbord), of
 - Touch screen
- *Geheugen*
 - 256 kilobytes non-volatile geheugen voor de MIDP



AFBEELDING 2. Overzicht van de lifecycle van een MIDlet

- omgeving, bovenop de eisen voor CLDC
- 8 kilobytes non-volatile geheugen voor persistente applicatie data
- 128 kilobytes volatile geheugen voor de Java runtime omgeving (de Java heap)
- *Netwerkverbinding*
 - Tweerichtingsverkeer
 - Draadloos
 - Mag af en toe onderbroken worden
 - Beperkte bandbreedte
- *Geluid*
 - De mogelijkheid om tonen te spelen.

Java applicaties voor het MIDP profiel worden ook wel MIDlets genoemd. Een MIDlet kan worden vergeleken met een Java applet. Een MIDlet bestaat uit tenminste één subclass van de class `javax.microedition.midlet.MIDlet`. Deze class definieert een aantal lifecycle methodes die door het systeem worden aangeroepen om de applicatie te starten, pauzeren, en te stoppen. Afbeelding 2 geeft een overzicht van de lifecycle van een MIDlet en de methodes waarmee tussen de verschillende toestanden gewisseld kan worden.

Een groep MIDlets kan verzameld worden in een MIDlet suite. Alle MIDlets in een bepaalde suite worden bij elkaar verpakt en ze worden als één geheel geïnstalleerd. Een suite vormt een soort afgeschermd gebied binnen de JVM, vergelijkbaar met een J2SE classloader.

HELLO WORLD!

Geen artikel over dit onderwerp is compleet zonder een Hello World voorbeeld. Voorbeeld 1 bevat de source code voor een simpele MIDlet die de beroemde boodschap op het scherm tovert. Er wordt een schermvullend user interface component aangemaakt en er wordt een commando aan toegevoegd waarmee de applicatie afgesloten kan worden.

```

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

public class HelloMidlet extends MIDlet
    implements CommandListener {

    // Initialize the Midlet Display variable
    private Display midletDisplay;

    // Initialize a variable for the
    doneCommand
    private Command doneCommand;

    public HelloMidlet() {
        // Retrieve the display from the
        // static display object
        midletDisplay = Display.getDisplay(this);

        // Initialize the doneCommand
        doneCommand = new Command("DONE",

            Command.SCREEN, 1);
    }

    /**
     * Create the Hello Midlet World TextBox
     and
     * associate the exit command and listener
     */
    public void startApp() {
        // Create the TextBox containing the
        // "Hello Midlet World!!" message
        TextBox textBox = new TextBox("Hello
        Midlet",
            "Hello Midlet World!!", 256, 0);

        // Add the done Command to the TextBox
        textBox.addCommand(doneCommand);

        // Set the command listener for the
        textbox
        // to the current midlet
        textBox.setCommandListener(this);

        // Set the current display of the midlet
        // to the textBox screen
        midletDisplay.setCurrent(textBox);
    }

    /**
     * PauseApp is used to suspend background
     * activities and release resources on the
     * device when the midlet is not active.
     */
    public void pauseApp() {
    }

    /**
     * DestroyApp is used to stop background
     * activities and release resources on the
     * device when the midlet is at the end of
     its
     * life cycle.
     */
    public void destroyApp(boolean uncondition-
    al) {
    }

    /**
     * The commandAction method is implemented
     by
     * this midlet to satisfy the Command-
     Listener
     * interface and handle the done action.
     */
    public void commandAction(Command command,
        Displayable screen) {
        // If the command is the doneCommand
        if (command == doneCommand) {
            // Call the destroyApp method
            destroyApp(false);

            // Notify the midlet platform that
            // the midlet has completed
            notifyDestroyed();
        }
    }
}

```

VOORBEELD 1. Een Hello World MIDlet

De user interface van de MIDlet uit voorbeeld 1 ziet er uit als in afbeelding 3. Deze afbeelding maakt meteen duidelijk dat er in de MIDP specificatie geen enkele garantie wordt gegeven over hoe de applicatie er precies uit gaat zien op een bepaald apparaat. Elke implementatie van MIDP toont de MIDP user interface classes en de commando knoppen met de standaard look en feel van dat apparaat. Hier moet flink rekening mee worden gehouden tijdens development. Het is dan ook aan te bevelen om een applicatie op zoveel mogelijk verschillende apparaten te testen.

Er zijn drie belangrijke toepassingsgebieden voor MIDP applicaties:

- Snelle toegang tot algemene informatie (het opzoeken van een goed restaurant in de buurt)
- Zakelijke toepassingen (het invullen van timesheets, het doen van bestellingen, logistiek)
- Vermaak (spelletjes)

USER INTERFACE

Er zijn drie verschillende typen schermen in de MIDP GUI:

- Schermen die volledig een complexe GUI bevatten,

zoals een Lijst of een TextBox component. Het structuur van deze schermen staat vooraf vast en de applicatie kan geen andere componenten toevoegen aan deze schermen.

- Generieke schermen die een Form component gebruiken. Deze applicaties kunnen tekst, plaatjes en een kleine set samenhangende UI componenten toevoegen aan deze Form, die dan als een container fungeert.
- Schermen in de betekenis van low-level API zoals een subklasse van de Canvas of een Grafische klasse.

OPTIONAL PACKAGES Naast configuraties en profielen is er nog een derde concept dat veel wordt gebruikt in J2ME, het *optional package*. Een optional package is een API die een zeer specifieke set functionaliteit biedt. Optional packages hoeven niet gebonden te zijn aan één configuratie of profiel. Soms hebben ze een heel breed toepassingsgebied, en soms zijn het uitbreidingen op een specifiek profiel. Tabel 1 geeft een kort overzicht van een aantal optionele packages en hun toepassingsgebied. Sommige van deze packages zijn nog in ontwikkeling.

CONCLUSIE

J2ME/MIDP is een veelbelovend, krachtig platform voor het bouwen van applicaties en games voor mobiele telefoons en andere handheld devices. De integratie met J2EE en webservices is prima en opent een nieuw client platform voor J2EE. De beschikbaarheid van apparaten met ondersteuning voor MIDP is de laatste paar jaar sterk gegroeid en hoeft niet langer een onzekere factor te zijn bij de keuze voor een ontwikkelplatform. Bij het ontwikkelen van applicaties met behulp van MIDP zijn er wel een paar valkuilen. Ook al zijn API's gestandaardiseerd in de vorm van configuraties en profielen, er kunnen wel degelijk een aantal verschillen in implementatie optreden tussen de daadwerkelijk apparaten waarop gedeployed moet



AFBEELDING 3. Het resultaat van voorbeeld 1 op twee verschillende apparaten

Naam	JSR	Korte beschrijving
Mobile Media API (MMAPI)	135	Een API die toegang biedt tot de audio en multimedia capaciteiten van mobiele apparaten, zoals de camera op een mobiele telefoon.
Wireless Messaging API (WMA)	120	Een API voor het gebruik van SMS en andere protocollen voor mobiele communicatie.
J2ME Web Services	172	Dit package maakt het mogelijk om web services aan te roepen vanaf alle J2ME configuraties
PDA API optional package for J2ME	75	Een combo van twee packages. Het PIM package biedt een API om toegang te krijgen tot Personal Information Management data zoals het adresboek, de agenda, en de to-do lijsten. Het File-Connection package bevat een API.
Mobile 3D Graphics API J2ME	184	Een API voor het produceren van geavanceerde 3D graphics op apparaten zonder 3D hardware ondersteuning of floating point processors.

TABEL 1: Een paar interessante J2ME optional Packages

worden. Test de user interface en performance van je MIDP software dus altijd op meerdere apparaten.

Het MIDP-profiel is speciaal ontwikkeld met het oog op mobiele telefoons en vergelijkbare apparaten

BRONVERWIJZINGEN

1. "J2ME Building Blocks for Mobile Devices, White Paper on KVM and the Connected, Limited Device Configuration (CLDC)". Sun Microsystems.
2. "Device programming with MIDP, Part 1", http://www.javaworld.com/javaworld/jw-01-2001/jw-0105-midp_p.html
3. White paper J2ME: "Programmeren in een mobiele wereld. Publicaties van de ervaringen" door diverse auteurs van Wisdom Assen BV, datum: november 2001; www.wisdom.nl

Age Mooy is senior Java architect en consultant (e-mail: amoooy@wisdom.nl). Mw. ir. Sherry Bibiana is chief technology officer. Beiden zijn werkzaam bij Wisdom Information Consultants.