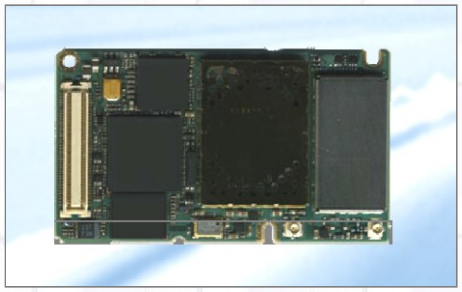


MCS M2M Java training

Siemens Wireless Modules
J2ME Authorized Training Center

Timo Medvedev (timo@mcs-nl.com)

Training overview

- Java History and Introduction
- Java crash course
- Java software & Siemens module JDK installation
- Application programming
- Autostart/OTAP
- Questions & discussion future application

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

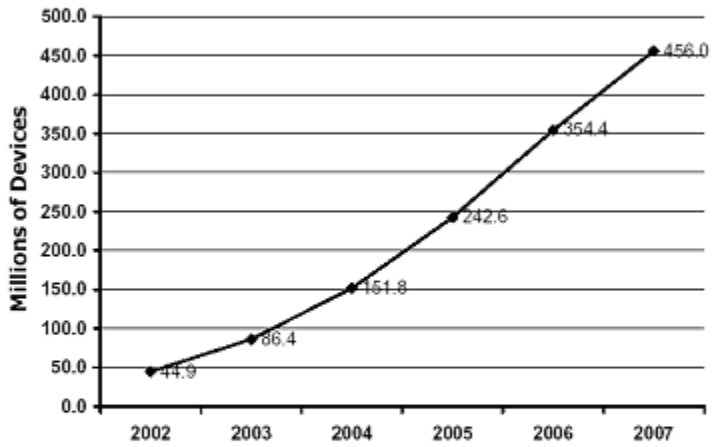
Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Java History – Past, Present, Future

- ✓ Birth December, 1990. The original name was Oak.
- ✓ April, 1995: Netscape announces Java embedded browsers, Java released to the masses.
- ✓ 2000: Java is used as e-business platform for secuire transactions.
- ✓ 2002-now: Java grows exponentialy in mobile market, see the chart on

Wireless Java Market Projections

Figure 3 Global Shipments Java Handsets (2002 - 2007)



	2002	2003	2004	2005	2006	2007
Total Handsets (Mil.)	401.8	450.8	509.1	550.4	584.7	613.7
Java Handsets (Mil.)	44.9	86.4	151.8	242.6	354.4	456.0
Java Handset Percent	11%	19%	30%	44%	61%	74%

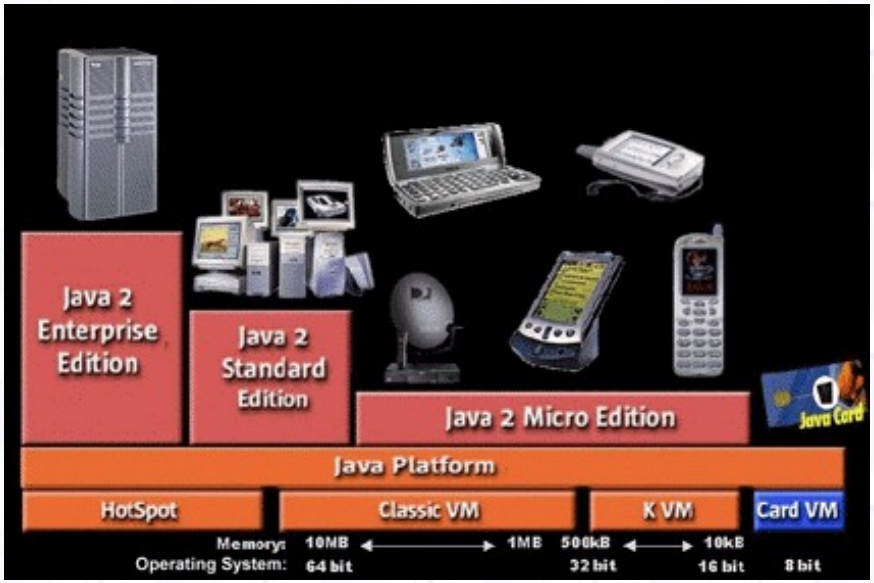
Source: Zelos Group

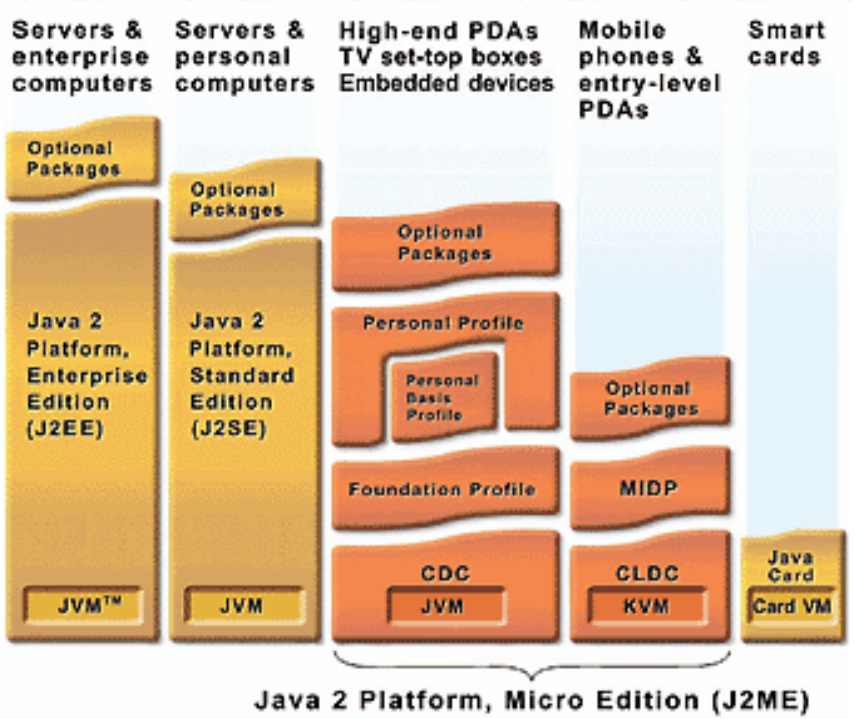
Java Introduction – What is JAVA?

- ✓ Programming Language
- ✓ Platform independent: Write once, run anywhere (on different platforms, Windows, Unix, Linux, etc..)
- ✓ Java Virtual Machine (JVM) - Runs Java programs
- ✓ 66 Java Technologies and 50 + Java Technologies in development
- ✓ Open Architecture license
  - ➢ Free usage, free IDE's available

Java Introduction – platform and technology

- ✓ J2EE – Java for server applications (backoffice)
- ✓ J2SE – Java for desktop (PC applications)
- ✓ J2ME – Java for mobile handsets (Phones, PDA's, washing machine…





Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Java Introduction – platform advantages

- **Cost**
  - ✓ Java is free regardless of total number of users
  - ✓ Free tools: Editors + Enterprise Servers (Jboss, Tomcat)
  - ✓ Free tools + free language => more free tools
- **Portability**
  - ✓ Write code once and run everywhere
- **Re-usability**
  - – Using past code in current problems.
  - – Some one has done your work. Use other's work.

Java Introduction – JAVA v.s. C++

- ✓ Java is becoming more and more popular in embedded device programming.
- ✓ Portability of Java code, write once and run everywhere (plaform independent)!
- ✓ Java Virtual Machine is handling memory management and garbage collecting.
- ✓ Java Virtual Machine prevents direct access to memory and resources,‘sandbox’.
- ✓ Less technical knowledge is needed when programming with Java v.s. C++

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Training overview

- Java History and Introduction
- **Java crash course**
- Java software & Siemens module JDK installation
- Application programming
- Autostart/OTAP
- Questions & discussion future application

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Java course - Basics (general)

- Java is case- sensitive:
  - MyVar is different to myVar!
- Spaces are not allowed in variables:
  - My Var is illegal variable name!
- Each statement has to finish with ";":
  - MyVar = 150;
- Each function has to have parenthesis:
  - getTime(), add(parameter)
- Boolean values are only true and false, you cannot use 1 and 0

Java course - Basics (language types)

- Primitive Types
  - float / double: 1.43, 3.6E18, -14.0
  - int: 12, 16384, -5000
  - boolean: true or false
  - byte – represents signed byte as integer between: -127 … 0 .. 128
  - char – represents characters in java: a, b, c, 1, 2, 3, ., :, “.
- Compound Types – Classes:
  - String – represents textual strings, e. g. “MSFT”, “Hi there!”, etc
  - ATCommand – is used for working with AT interface on TC65

Java course - Basics (Operators)

- TypesArithmetic operations
  - Notation: + , – , * , / , % (remainder)
  - In integer divisions, the fraction part is lost, e. g. 3 / 2 equals 1, and 2 / 3 equals 0
  - The ‘ + ‘ operator allows operands of type String (concatanating Strings)

- Comparison operations
  - Notation: > , >= , < , <= , == (equal) , != (unequal)
- Boolean operations
  - Notation: && (AND), || (OR), ! (NOT)
- Conditional operator
  - Notation: condition ? value- if- true : value- if- false (short notation for if-else)
- Assignments and shortcuts
  - Notation: = , += , -= , *= , /= , %= , ++ , --
  - Example: a+= b is equivalent to a= a+ b

Java course - Basics (classes and objects)

- Most classes require to create an object of their type:
  ClassName variable = new ClassName();

- or in two lines:
  ClassName variable = null;
  variable = new ClassName();

- or if class has constructor which requires parameters:
  ClassName variable = new ClassName(parameter1, parameter2);

- When classes have static method than it is no need to create an object.
  ClassName.someMethod();

  Example: ***System.out.print("Hello world");***
  Calendar mijn = Calendar.getInstance();

Java course – Control structures (1)

- **IF / ELSE**

```
if ( a > b ) {   System.out.println(a); }
else {    System.out.println(b); }
```

- **SWITCH / CASE**

```
switch ( n )
{
   case 1: System.out.println("one");
             break;
   case 2: System.out.println("two");
              break;
   default: System.out.println(“nothing selected");
}
```

Java course – Loop structures (2)

- **FOR**

```
for ( int i=0; i<n; i++ )
{
  System.out.println( i );
}
```

- **REVERSED FOR**

```
for ( int i=n-1; i>=0; i-- )
{
  System.out.println( i );
}
```

- **DUAL FOR**

```
for ( int i = 0 ,j = 0; i<n;
i++ ,j ++ )
{
  System.out.println( i +" :
"+ j );
}
```

- **WHILE**

```
while ( moreData() )
{
   readIt(byte[] data);
}
```

- **DO / WHILE**

```
do
{
  readIt();
  if ( done() ) break;
  if ( bypassThisOne() )
continue;
}
```

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)l
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Java course – Inheritance

- Inheritance: mechanism for extending behavior of classes; leads to construction of hierarchy of classes.
- What happens when class B extends class D:
  - Inherits instance variables
  - Inherits static variables
  - Inherits instance methods
  - Inherits static methods
  - B can:
    - Add new instance variables
    - Add new methods (static and dynamic)
    - Modify methods (only implementation)
    - Cannot delete anything

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Java course – Inheritance (example)

```
public class ComPort() {
    public byte[] read()                {       // read data from COM port     }
    public void write(byte[] data)      {       // write data to COM port      }
}

public class GpsReader extends ComPort  {
    public String getPosition() {
            byte[] data = super.read();
            // use functionality of ComPort to read data and do high level processing here
    }
}

public class DeviceReader extends ComPort {
    public String getMeterData(int meterNumber) {
            byte[] data = super.read();
            // use functionality of ComPort to read data and do high level processing here
    }

    public void setMeterConfiguration(String config)  {
            // use functionality of ComPort to write data and do high level processing here
            super.write(data);
    }
}
```

Java course – Threads

- **Threads in Java**
  - ✓ Java is multitasking oriented!
  - ✓ Java manages and schedules threads
  - ✓ Java provides “synchronize” to help coordinate multiple threads and access to data

Java course – Threads (example)

```
public class MyThread extends Thread {
    public MyThread(String threadName) {
        super(threadName);
    }

    public void run() {
        // Execute some task here....
    }
}

public class ThreadTest {
    public static void main(String[] args)
    {
        MyThread t = new MyThread(args[i]);
        t.start();
    }
}
```

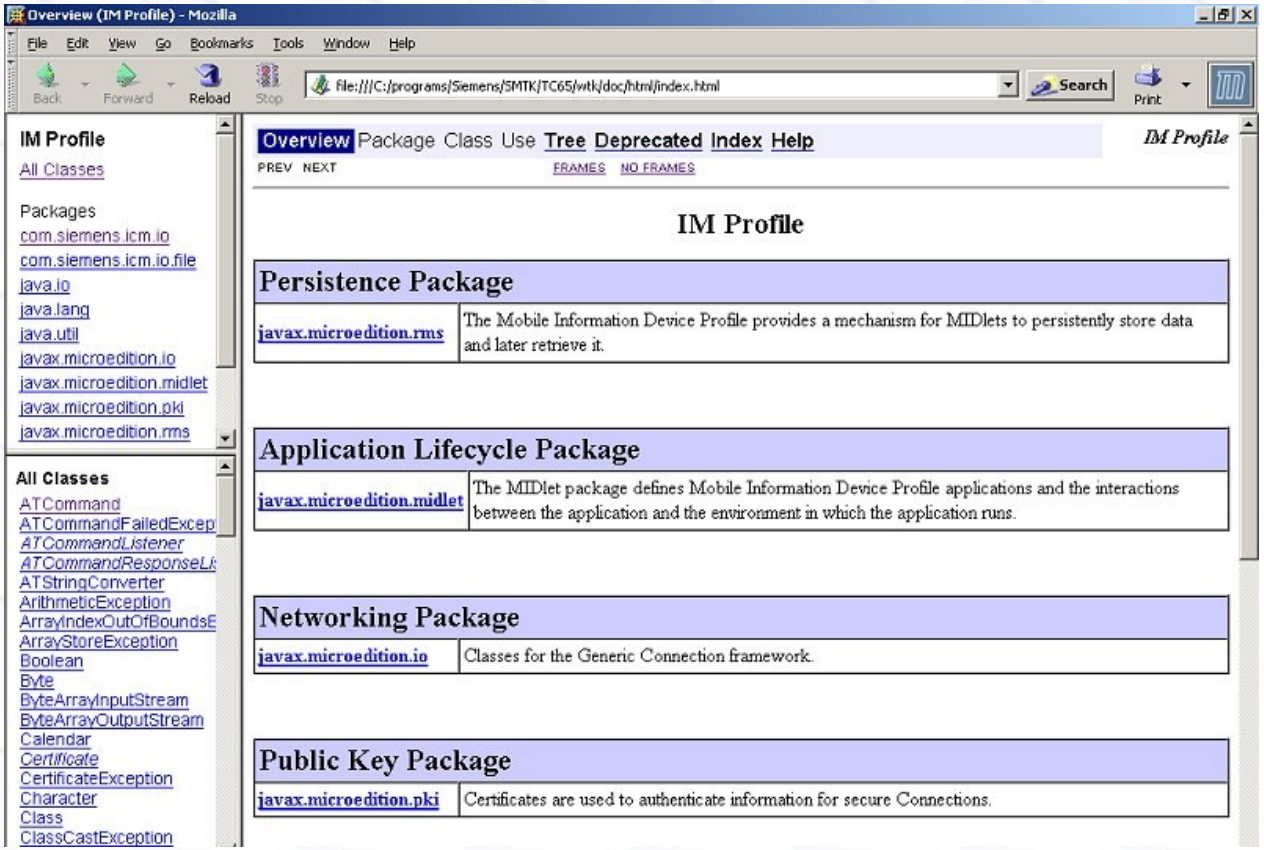
Java course – Handling exceptions/errors

```
public void processData()
{
    try
    {
        readDataFrom(file);   // this code might throw an Exception
    }
    catch ( IOException e )
    {
      System.out.println(e.getMessage());
      // handle the situation;
    }
    finally
    {
        file.close(); // always executed
    }
}
```

Java course - Classes documentation

All Java classes are documented in so called javadocs style.

Siemens module classes and methods are documented in HTML pages, see <YourDir>/wtk/docs/



Java course – Sources of information

Siemens modules API Specification:

- <YourDir>/wtk/docs/index.html

Demo

Books on Java and J2ME:

- J2ME in a Nutshell (by Kim Kepley)
- Wireless Java Programming with J2ME (Dr. Jun Zhu)

Online information sources:

- MCS M2M forum (www.mcs-nl.com/support)
- Siemens Mobile forum (www.benqmobile.com/developer)

Training overview

- Java History and Introduction
- Java crash course
- **Java software & Siemens module JDK installation**
- Application programming
- Autostart/OTAP
- Questions & discussion future application

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

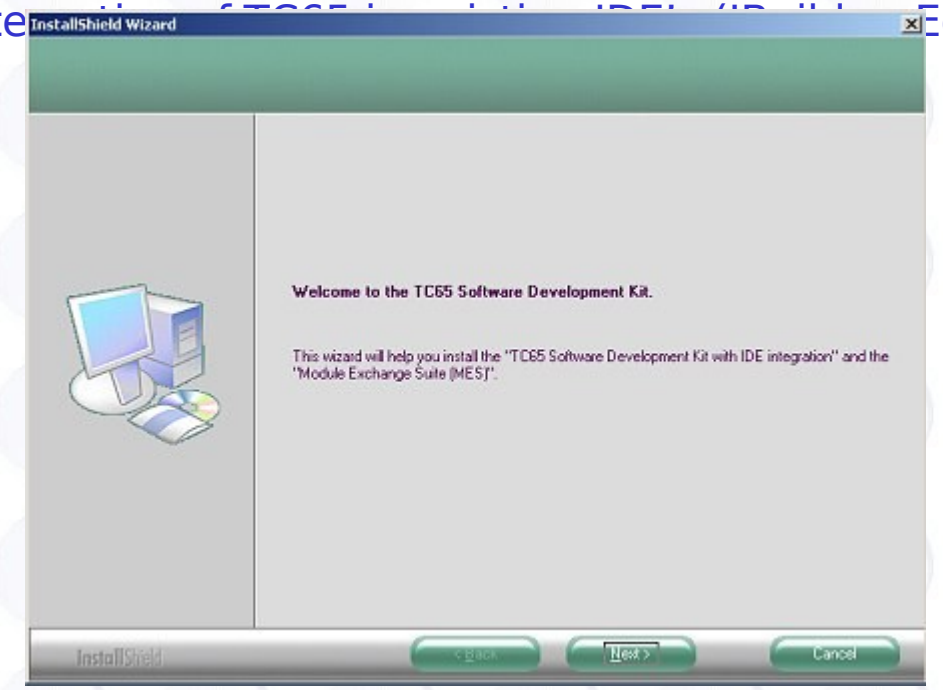
Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Installation – Java & Eclipse IDE (1)

- ✓ Run from CD
  \...\JDK 1.4\j2sdk-1_4_2_09-windows-i586-p.exe
- ✓ Copy Eclipse archive from CD
  \...\Eclipse\eclipse-sdk-3.1.2-win32.zip
  to C:\programs\ on your harddrive.
- ✓ Unzip archive, this will put Eclipse in
  C:\programs\eclipse
- ✓ Start Eclipse b  ing
  :\programs\ed clipse.exe

Installation – Siemens SDK Toolkit CD

- ✓ Start setup.exe in SDK toolkit directory on CD
- ✓ The setup software will install:
  - ✓Java SDK environment version 1.4.2_07 if needed
  - ✓Sun One Studio Mobility IDE if chosen (Eclipse needs to be installed manually!)
  - ✓MES software is needed
  - ✓Setup inte[illegible]Eclipse, Sun One Studio).

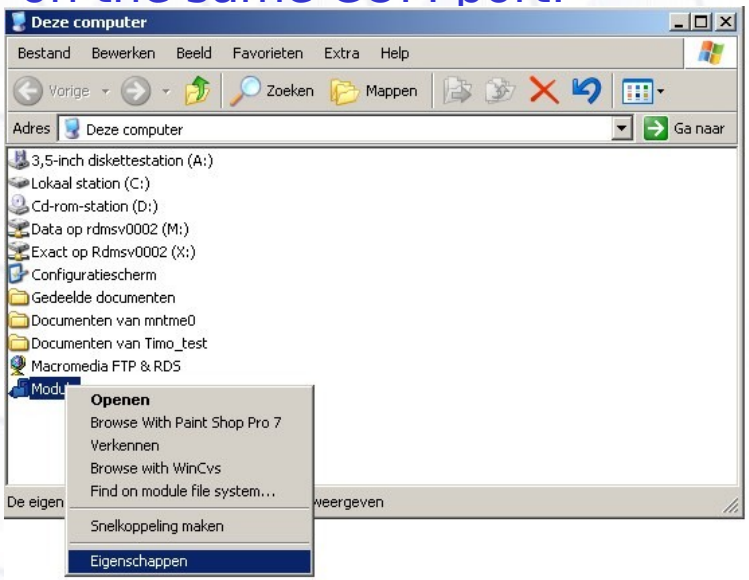


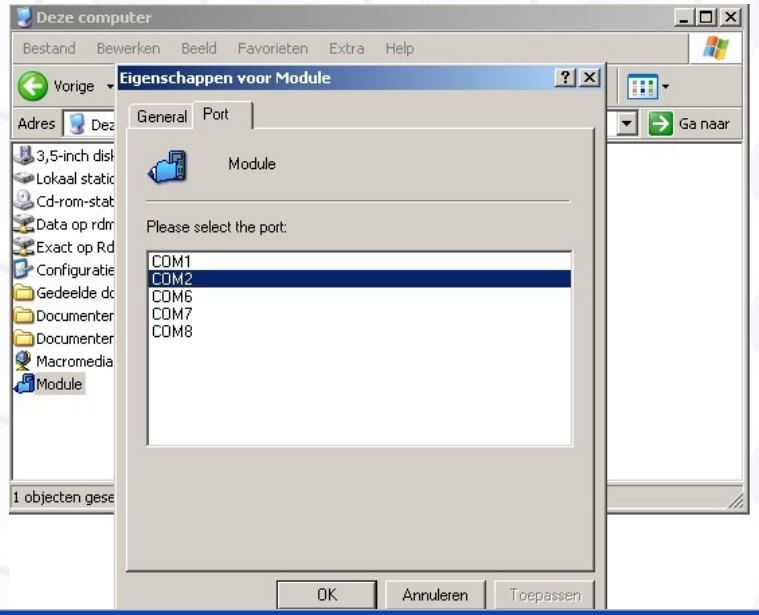
Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Installation – Module Exchange Suite

- ✓ View the Flash file system from Windows Explorer.
- ✓ Make sure that the module is connected to the COM port that the Module Exchange Suite is configured to.
- ✓ Module should be in normal mode and and hyperterminal is not active on the same COM port.





Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

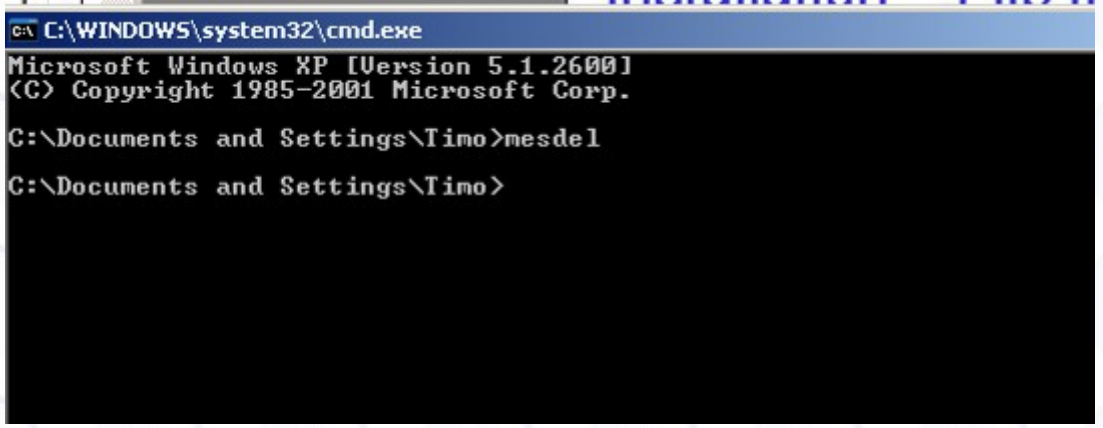
Installation – File transfer to module

Windows Based

To transfer a file to the module, simply use your Windows Explorer.

Command Line Based

A suite of command line tools is available for accessing the module's Flash file system. MESdel, MEScopy, MESxcopy, MESdir, MESmkdir, MESrmdir, MESport, MESclose and MESformat.



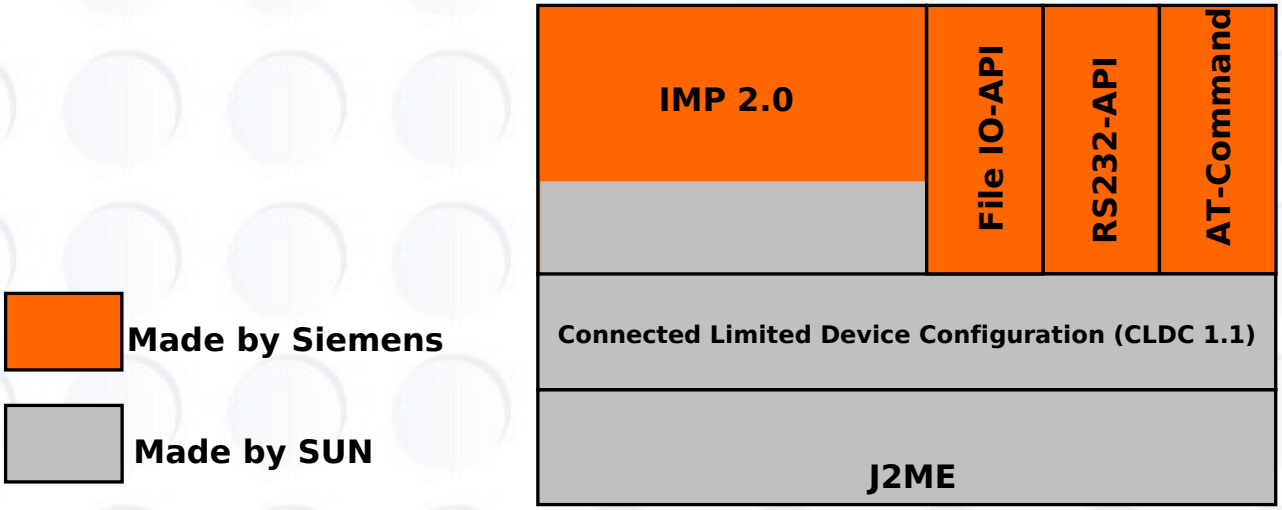
Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Programming – Platform architecture

Software Architecture:

- Based on Java Platform Micro Edition (J2ME)
- Provided by SUN Microsystems Inc.: http://java.sun.com/j2me/
- Information Module Profile (IMP-NG)
- Device specific platform for modules
- IMP-NG = MIDP 2.0 – LCDUI (= Standard MIDP without grafical Interface)

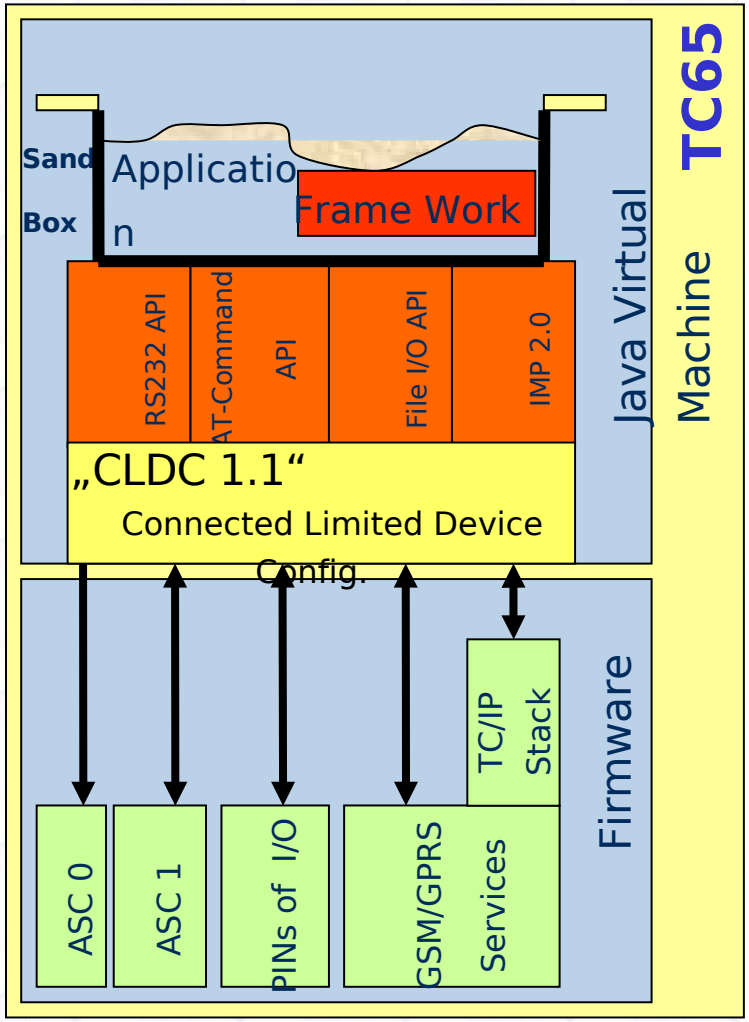


Programming – Sand Box Model

Application runs in „Sand Box“.
This
means that access to module‘s resource is only possible through provided APIs.

APIs:

- ✓ (Application Program Interface)
  - ✓RS232 API
  - ✓AT-Command API
  - ✓File I/O API
  - ✓IMP 2.0



Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Programming - MIDlet application

- The J2ME™ Mobile Information Device Profile (MIDP) provides a targeted Java API for writing wireless applications.
- MIDP applications are referred to as MIDlets. MIDlets are controlled by the mobile device implementation that supports the CLDC and MIDP. Since IMP-NG is a subset of MIDP 2.0, IMP includes MIDlets.
- The MIDlet class in the MIDlet package provides methods to manage a MIDlet's life cycle.

Programming - MIDlet application life cycle

The MIDlet life cycle defines the protocol between a MIDlet and its environment.

A MIDlet has three valid states:

- ✓ Paused – The MIDlet is initialised and is quiescent. (It should not be holding or using any shared resources).
- ✓ Active – The MIDlet is functioning normally.
- ✓ Destroyed – The MIDlet has released all of its resources and terminated. This state is only entered once.

Siemens presentation

Programming - MIDlet application structure

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

public class HelloWorld extends MIDlet
{
  public HelloWorld()                    {              }

  public void startApp() throws MIDletStateChangeException
  {
    // put here your application code
    destroyApp(true);
  }

  public void pauseApp()                          {              }

  public void destroyApp(boolean cond)  { notifyDestroyed();}
}
```

Programming – Application (executable)

- JAR → Java Archive
- The JAR file is a compressed file, which contains the java program (MIDlet + all the other Classes and resources created by the user).
- JAD →Java Application Descriptor
- The JAD file describes the JAR file and its properties (name, size, version, application parameters…)
- Besides class files JAR can contain other resources (images, data files, configuration files).

Eclipse Demo

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Training overview

- Java History and Introduction
- Java crash course
- Java software & Siemens JDK installation
- **Autostart/OTAP**
- Questions & discussion future application

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

Autostart – Application automatic start

Most applications are required to autostart when module is powered up:

- ✓ Setup appliaction name:
  AT^SCFG="Userware/Autostart/AppName","<password>","a:/<dir>/<app.jad>"
- ✓ Setup application autostart delay (in steps of 100ms):
  AT^SCFG="Userware/Autostart/Delay"," <password>","100"
  (start application 10 seconds after power up)
- ✓ Setup application autostart ON/OFF:
  AT^SCFG="Userware/Autostart"," <password>","1" (Use 0 to switch autostart off)

Autostart – Switch off automatic start

There are two methods for switching off the autostart feature:

- the at^scfg AT command, or
- the “autostart_off.exe” tool (included in the Installation CD software under wtk/bin)

To disable the automatic start of a user application follow these steps:

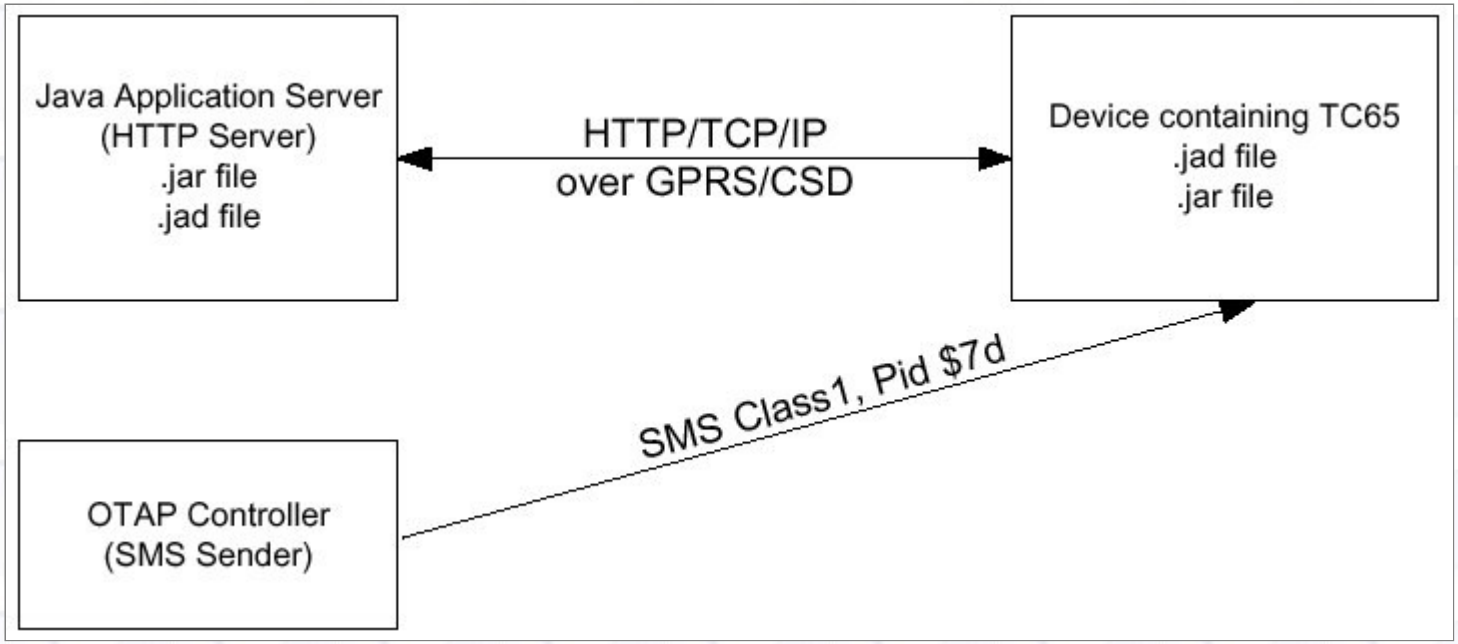
1. Connect the module to the PC
2. Make sure, that the module is switched off
3. Start the Autostart_Off program
4. Select the COM-Port
5. Press the “Autostart Off” button

OTAP – Over The Air Provisioning overview

- OTAP is a mechanism to install, update and delete JAVA applications over the air.
- OTAP is a common practice in the Java world
- OTAP can be fully controlled over SMS and also via AT commands

Requirements for OTAP:

- Web server wich contains .jad and .jar files
- OTAP Controller or SMS Sender



OTAP – Over The Air Provisioning (AT)

To initiate OTAP with AT command use the following steps:

- ✓ Setup OTAP parameters:
  AT^SJOTAP=”<password>”,http://www.someserver.com/app.jad,”a:/<dir>/”,”<http_user>”, “http_password”, “<gprs/csd>”, “<APN/Tel.number>” [“gprs_user”,”gprs_password”]

- ✓ Initiate OTAP:
  AT^SJOTAP + [enter]

- ✓ Trace/Log OTAP process:
  AT^SCFG="Trace/Syslog/OTAP","1“ (Notice: the module won’t except any AT commands after this command is performed).

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

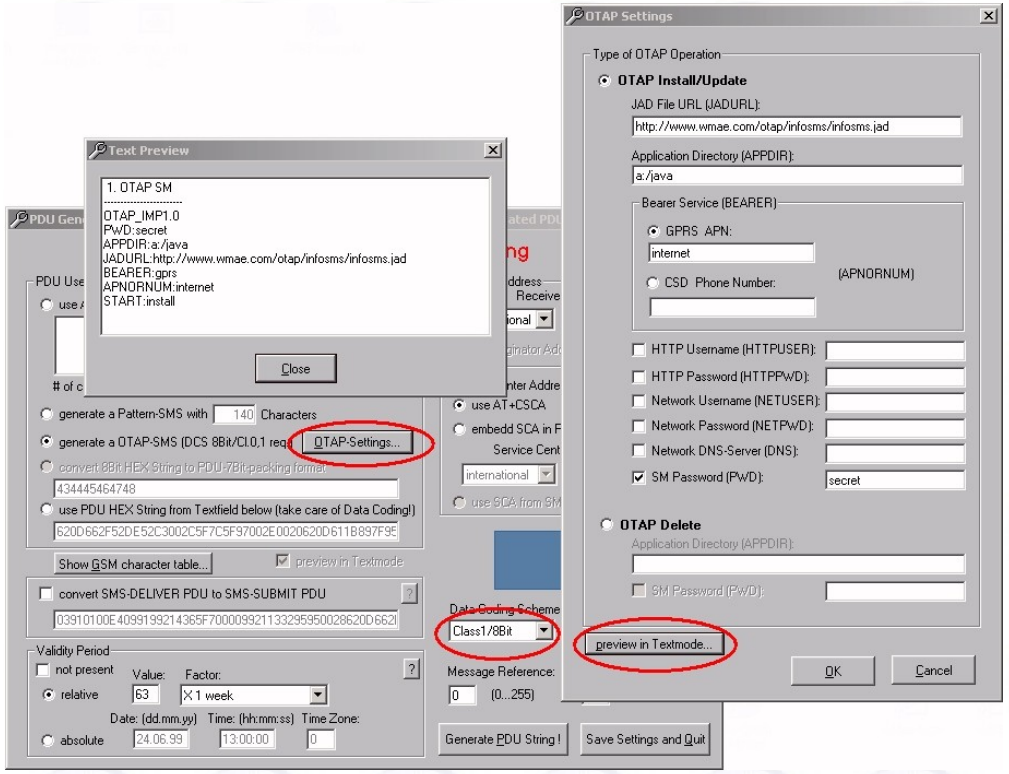
Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

OTAP – Over The Air Provisioning (SMS) 1

To initiate OTAP with SMS command (somewhat more complicated):

Two types of OTAP operations:

- Install/Update
- Delete



PDU Generator

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com

OTAP – Over The Air Provisioning (SMS) 2

The first line is required: it is used to identify an OTAP SM. All other lines are optional and their order is insignificant, each line is terminated with an LF: '\n'

including the last one.

OTAP install/update SMS:

```
OTAP_IMPNG
PWD:secret
JADURL:http://www.some.com/mega.jad
APPDIR:a:/work/appdir
HTTPUSER:user
HTTPPWD:anothersecret
BEARER:gprs
APNORNUM:access.to-thenet.net
NETUSER:nobody
NETPWD:nothing
START:install
```

OTAP delete SMS:

```
OTAP_IMPNG
PWD:secret
APPDIR:a:/work/appdir
START:delete
```

OTAP – Trace OTAP procedure

For easy debugging of the OTAP scenario, the OTAP procedure can be traced over the serial interface.

- **AT^SCFG=Trace/Syslog/OTAP,1**

Questions & Discussion

Questions?

Customer application discussion?

Kiotoweg 719-721, NL- 3047 BG Rotterdam
Tel: +31-10-4375555 Fax: +31-10-4375656
www.mcs-nl.com info@mcs-nl.com

Pegasuslaan 5, B-1831 Diegem (Brussel)
Tel: +32-2-7164837 Fax: +32-2-7164108
www.mcs-be.com info@mcs-be.com