
- Many features from script languages such as Python, Ruby and Smalltalk
- Become successfully popular
  - Supports all Java APIs classes and libraries
  - Dynamically compiled into the Java bytecode-executable on the Java Virtual Machine
- Script languages become popular in building graphical user interfaces and web applications.
- Write code easily to do complex
- Fast prototyping by using compact codes
- Longer execution time
- Unambiguous grammars
- Insecure type that might cause runtime errors

While overcoming all disadvantages, Groovy becomes commended in many ways building several applications successfully.

As the Java Specification Requests (JSR) process in the Java Community Process (JCP), the Groovy project delivers three key artifacts:

- Groovy Language Specification (GLS)
  - Syntax and semantics
- Reference Implementation (RI)
  - Binary distribution of the Groovy scripting language which passes TCK
- Test Compatibility Kit (TCK)
  - A suite of tests, tools and documentation that determines whether or not a Groovy implementation (RI or third-party implementation) complies with the GLS

What is Groovy?

- Groovy is an agile dynamic language for the Java 2 platform.
- The basic features of Groovy are from other languages such as Java, Python, Ruby and Smalltalk.
- Groovy brings the power of a scripting language into the Java 2 platform.
Groovy Community

- Contribution to Groovy is appreciated to improve the Groovy language to better components.
- Maintained with Wiki
- Mailing lists
- Located at the Codehaus that is an extension of the Werken Company that provides infrastructure for open-source projects
- The Groovy community
  - 23 members who directly contribute to the Groovy project
  - Members have the rights to directly access the source of a project and actively evolve the code base.
  - 19 contributors who provide valuable suggestions
  - Contributors support the projects by giving patches or suggestions to the members.

The Groovy language is submitted as a Java Specification Requests (JSR):
- “JSR 241: The Groovy Programming Language”.
- Driven by a Specification Lead, James Strachan, and 17 Expert Group members such as Aaron Alpar, Apache Software Foundation, Boeing, IBM, Guillaume Laforge, Richard Monson-Haefel, Sun Microsystems, and etc.

Getting Started
- Install Groovy
  - http://groovy.codehaus.org/Download

Groovy Interface for Ubiquitous Computing
- There are limitations to support applications on various ubiquitous computing devices.
  - They use different types of hardware and they need to support various interfaces while consistently supports running various applications and communications.

Java 2 Platform Micro Edition (J2ME)
- To support commodity in the various consumer devices such as mobile phones, personal digital assistants (PDAs), pagers, set-top boxes, and vehicle telematics systems

<table>
<thead>
<tr>
<th>Device Family</th>
<th>Special Packages</th>
<th>Optional Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Java 2 Platform Micro Edition (J2ME)
- Java provides the Java 2 Platform Micro Edition (J2ME) to support commodity in the various consumer devices such as mobile phones, personal digital assistants (PDAs), pagers, set-top boxes, and vehicle telematics systems (Java 2005). J2ME is a set of technologies and specifications developed for mobile devices that have limited resources such as limited interface and limited power. J2ME uses subset components of Java 2 Platform Standard Edition (J2SE) such as smaller virtual machines and leaner APIs.
Various J2ME technologies have been developed and the associated specifications are defined. Currently they include:

- Connection Limited Device Configuration (CLDC)
- Mobile Information Device Profile (MIDP)
- Java Technology for the Wireless Industry (JTWI)
- Mobile Media API (MMAPI)
- Location API for J2ME
- Security and Trust Services API for J2ME
- MIDP Web Services APIs
- Bluetooth API
- J2ME Content Handler API

Connected Device Configuration (CDC)
- Foundation Profile (FP)
- Personal Basis Profile (PBP)
- Personal Profile (PP)
- J2ME Content Handler API
- Personal Java Card
- Secure Transaction API
- Java Telephony API
- Java TV APIs
- Small-screen AUI
- Java Telephony Technology

J2ME has two main branches. The first branch is based on the Connected Limited Device Configuration (CLDC). This configuration is for small wireless devices such as mobile phones and PDAs. Based on CLDC, the Mobile Information Device Profile (MIDP) defines the higher level profile for the J2ME application environment. There are many MIDP-compliant devices [Figure 1] produced by consumer companies such as Alcatel, Casio, Fujitsu, Hitachi, Kyocera, LG, Mitsubishi, Motorola, NEC, Nokia, Panasonic, Samsung, Sharp, Siemens, Sony Ericsson, Toshiba, and etc [J2ME Devices 2005].

The other major branch of J2ME is based on the Connection Device Configuration (CDC). This configuration is for larger devices (in terms of memory and processing power) with more robust network connection such as set-top boxes and Internet appliances. The Foundation Profile extends CDC and serves as the basis for several other profiles. CDC is basically a superset of CLDC. It includes all the classes defined by CLDC, including new ones that are not included in J2SE. CDC includes many more core J2SE components than CLDC.

The current J2ME universe is depicted as follows [excerpted from Java 2005].

To use Groovy on mobile devices that allow Java bytecode classes, we may compile Groovy scripts and classes, and then upload to run them on the mobile devices. As long as the mobile device has enough space (1.5 MB) to hold the Groovy runtime, it may work [Laforge 2005].

The Groovy code base might use some APIs which might not be available on J2ME. Therefore, it will be probably a lot of work to trim down a new Groovy version for the mobile devices.
Acknowledgement

These notes are summarized mainly from the following references:

- Openwave Developer Network (ODN), 2006.