Apache DeviceMap

Werner Keil

@wernerkeil | wkeil@apache.org

APACHECON EUROPE

CORINTHIA HOTEL
BUDAPEST, HUNGARY
— NOVEMBER 17-21, 2014 —

Agenda

- Introduction
- History and Comparison
- Project Brief
- Tests and Results
- Examples and Demo
- Outlook
- Conclusion

Who am I?

- Consultant – Coach
- Creative Cosmopolitan
- Open Source Evangelist
- Software Architect
- JCP EC Member
- DevOps Guy …

Twitter @wernerkeil
Email wkeil@apache.org

apache devicemap

INTRODUCTION

Many, Many Mobile Devices

Presentation, Data Collection, Interaction and Utility

What is a DDR?

Deutsche Demokratische Republik (DDR)
7. Oktober 1949 3. Oktober 1990
Flagge der DDR
Wappen der DDR
FDJ
Freie Deutsche Jugend (FDJ)
SEID BEREIT
Jungpioniere
Erich Honecker
(Staatsratsvorsitzender)
Berlin Ost
stepmap.de

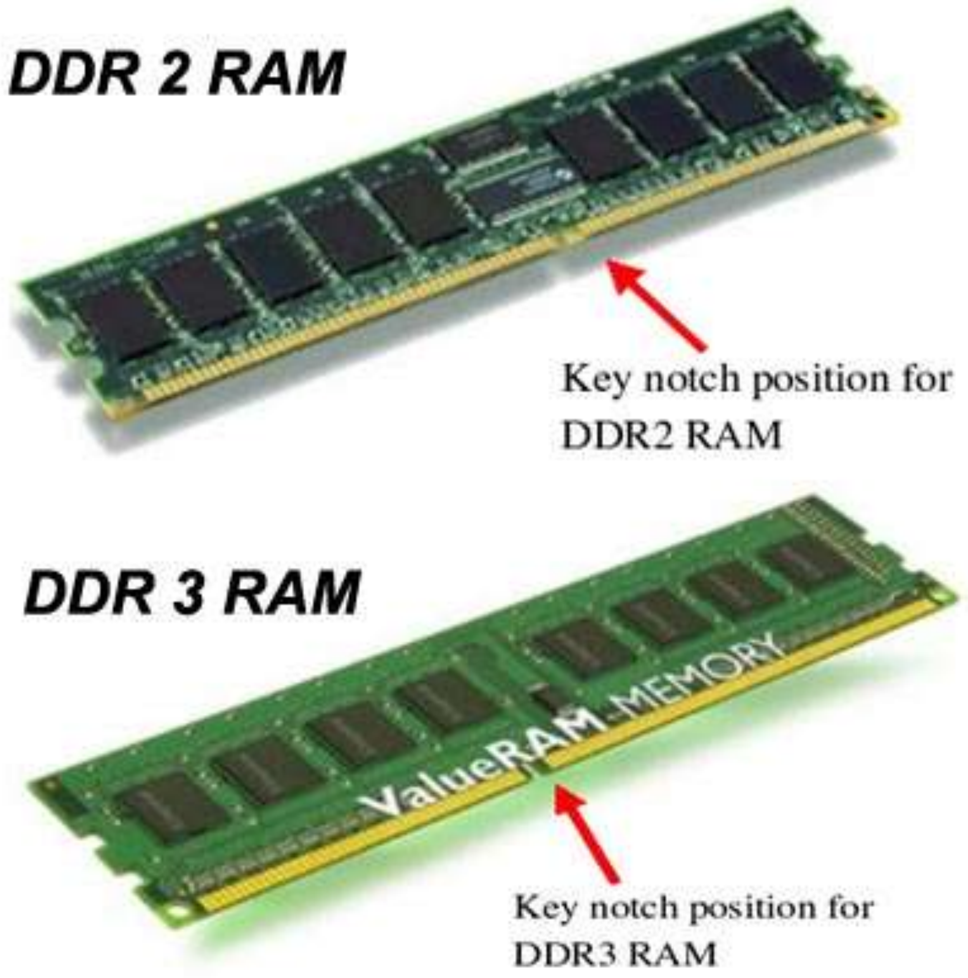
apache devicemap

What is a DDR?

What is a DDR?

apache devicemap

What is a DDR?


DDR 2 RAM
Key notch position for DDR2 RAM
DDR 3 RAM
ValueRAM MEMORY
Key notch position for DDR3 RAM

What is DeviceMap?

Day after day we all experience an incredible growth and fragmentation of devices available on a more and more proliferated market.
Accurately tracking their specs has become very hard work.

But this complexity can be reduced if managed by a community daily involved in improving the Device Description Repository (DDR).

This is the essence of the DeviceMap project, the best open and free repository of device descriptions currently available.

Why is DeviceMap different?

If you want a comfortable user experience, you need dynamic, “responsive” content according to hardware and browser of your device.

Device Description Repositories (DDR) are databases that plenty of information concerning mobile phones, tablets, Interactive TVs, set top boxes and any device having a Web browser. DDR allow developers to realize applications providing appropriate user experience on each client

There are several DDR projects, most demand often hefty fees to access to their databases and APIs or restricts free use to non-commercial applications.

HISTORY

WURFL Wall

AUFMARSCHIERT
Männer der paramilitärischen „Kampfgruppen der Arbeiterklasse" sperren am 13. August 1961 das Brandenburger Tor – angeblich, um den Frieden zu retten. Die DDR verwendet dieses Propagandabild jahrzehntelang, um den „antifaschistischen Schutzwall" zu rechtfertigen

WURFL Wall since August 2011

WURFL no longer Open Source

Project Comparison

Project	Strength	Weakness	License
MaDDR Project	The interface is a W3C standard	Device repository only with commercial mobileAware DDR (the APIs are bundled with a small sample DDR) The maDDR project cannot offer an adaptation technology that uses repository knowledge to guide its processes.	Repository: Commercial license only API: Commercial license OR Simple DDR API implementation licensed as LGPL.
DeviceAtlas	Data is multi-sourced from industry-leading partners. Some W3C compliance	Only commercial licenses.	Repository: Commercial license API: Commercial license
Volantis	Relatively wide device coverage	Only commercial licenses. Multiple takeovers, the new owner does not show support for it any more.	Repository: Commercial license API: Commercial license

Project Comparison (2)

Project	Strength	Weakness	License
WURFL	Former Community project (till Aug 2011)	The license does not allow to use the DDR without accessing through the WURFL API the API does not allow use in projects with a proprietary license	Repository: Use not allowed if accessed without WURFL APIs API license: dual licensing AGPL and commercial (Changed since August 2011)
51Degrees.mobi	More Predictable product catalogue, .NET Support	Free license contains fewer devices, usually less accurate	Repository: MPL or Commercial („Pro Edition“) API: Mozilla Public License
NetBiscuits	Service based, seems easy to integrate at least via JavaScript	Free and SME offerings dependent on using their JavaScript , no API below Enterprise or “OEM” price range	Repository: N/A API: Commercial license

PROJECT BRIEF

What was Contributed?

- In late 2011 a few months after OpenDDR started, Apache committers and members got aware of the technology and its relevance, too. Deciding to create both a unified Device Repository and APIs for common languages including Java.
- OpenDDR is a Founding Member of the DeviceMap Incubator. And made initial contribution of W3C DDR Libraries for Java and other languages as well as resource data.
- Also see the Proposal Wiki: http://wiki.apache.org/incubator/DeviceMapProposal

Metro – DeviceMap for .NET

apache devicemap

Why Open Standards?

- DeviceMap stakes on open standards:
  - Repository is fully compliant with W3C DDR
  - SimpleDDR library implements the W3C DDR Simple APIs
- From a developer point of view:
  - You can start developing your application with confidence that your product will work with any W3C DDR Simple API implementation, no binding to DeviceMap DDR implementation
  - Adopting a W3C standard, the Copyright of these interfaces is owned and protected by W3C against any IP or patent claim
- The Apache License means that you are free to use all of DeviceMap in open source or proprietary software.

What's different in DeviceMap?

- Users can update the operating systems of their devices either with custom builds, or just install a new browser.
- E.g. Google‘s Project Ara aims for modular devices with an almost endless combination of capabilities.
- The identification of a device through the original User Agent exposed by manufactures is no longer sufficient.
- DeviceMap considers the device as a combination of three important aspects:
  - Physical Device
  - Operating System
  - Web Browser

Level of Confidence

- DeviceMap can identify custom builds of the operating system and third party browsers.
- If a particular version of a browser or an operating system is not recognized, DeviceMap returns information about the closest version rather than not returning any information at all.
- DeviceMap identifies a device, a web browser or an operating system with a certain level of Confidence.
- You can specify the threshold as a target to achieve in the identification process. The higher the confidence the slower or more memory consuming the process may be.

Patching and Optimization

- Lowering the confidence speeds up the process, even if you can have less accurate results.
- DeviceMap supports Patching the data source.
- In DeviceMap we can specify a reduced vocabulary in order to limit the memory load.
  - It can be specified in the *oddr.limited.vocabulary.path* property of the DeviceMap properties file.
  - It allows developers to choose which properties need to be loaded in memory at startup from the datasource. In most case developers need only a subset of properties from the datasource.

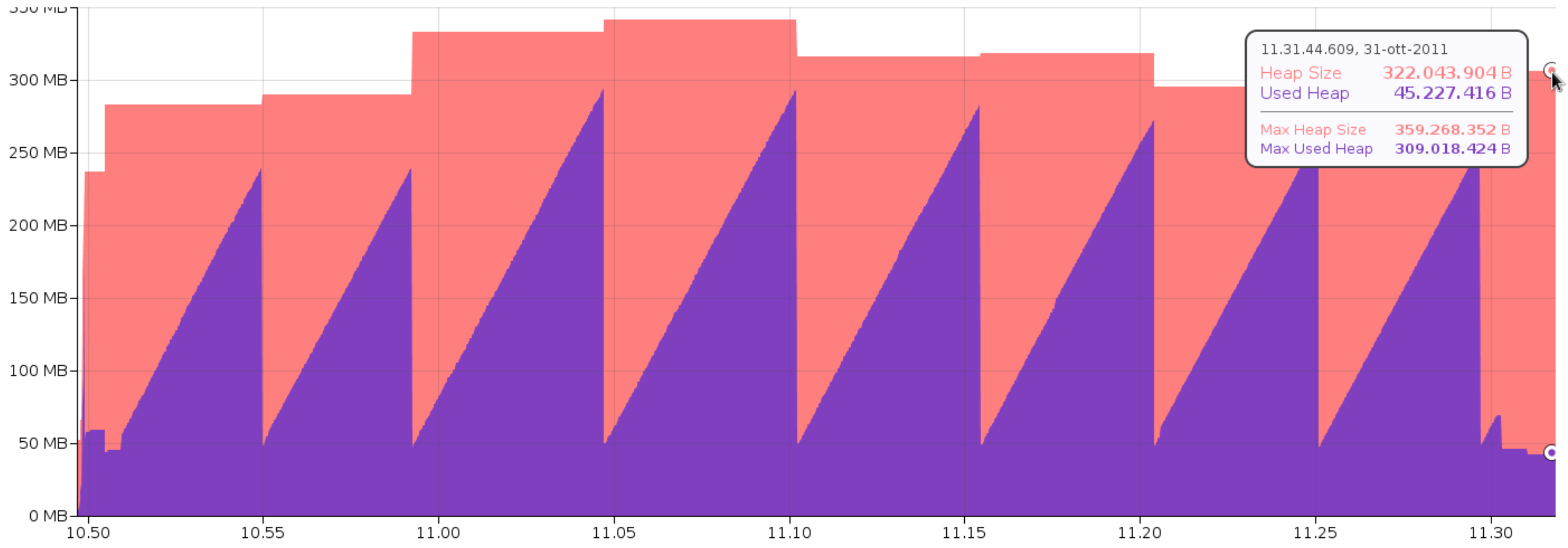
apache devicemap

TESTS

How we tested?

- We compared DeviceMap with WURFL API (1.3.1) in terms of memory load in two situations:
    - Without limited vocabulary
    - With limited vocabulary
- We used jMeter as workload generator, configuring it in order to generate 100 requests per seconds from 10 workers with a different user agent. The user agents set contained ten of most common user agent observed in a real environment.
- Both DeviceMap DDR and WURFL API were tested in a simple web app.

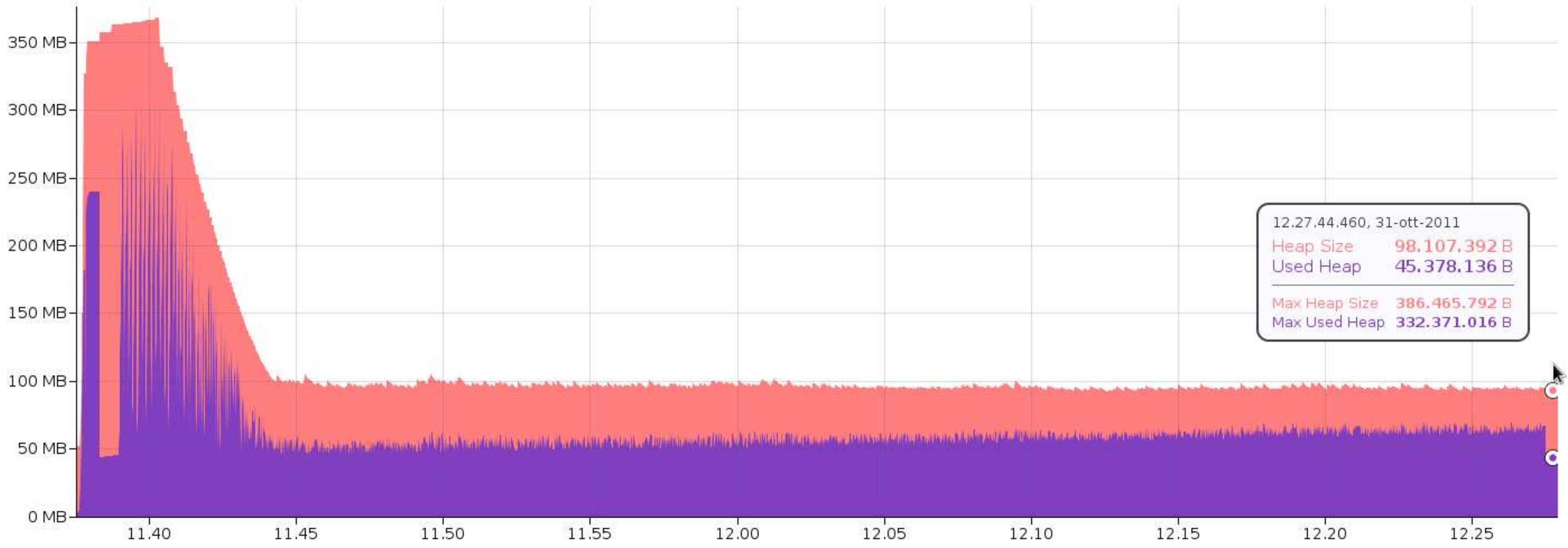
WURFL Memory Load



11.31.44.609, 31-ott-2011

Heap Size	322.043.904 B
Used Heap	45.227.416 B
Max Heap Size	359.268.352 B
Max Used Heap	309.018.424 B

Unreduced DeviceMap Memory Load

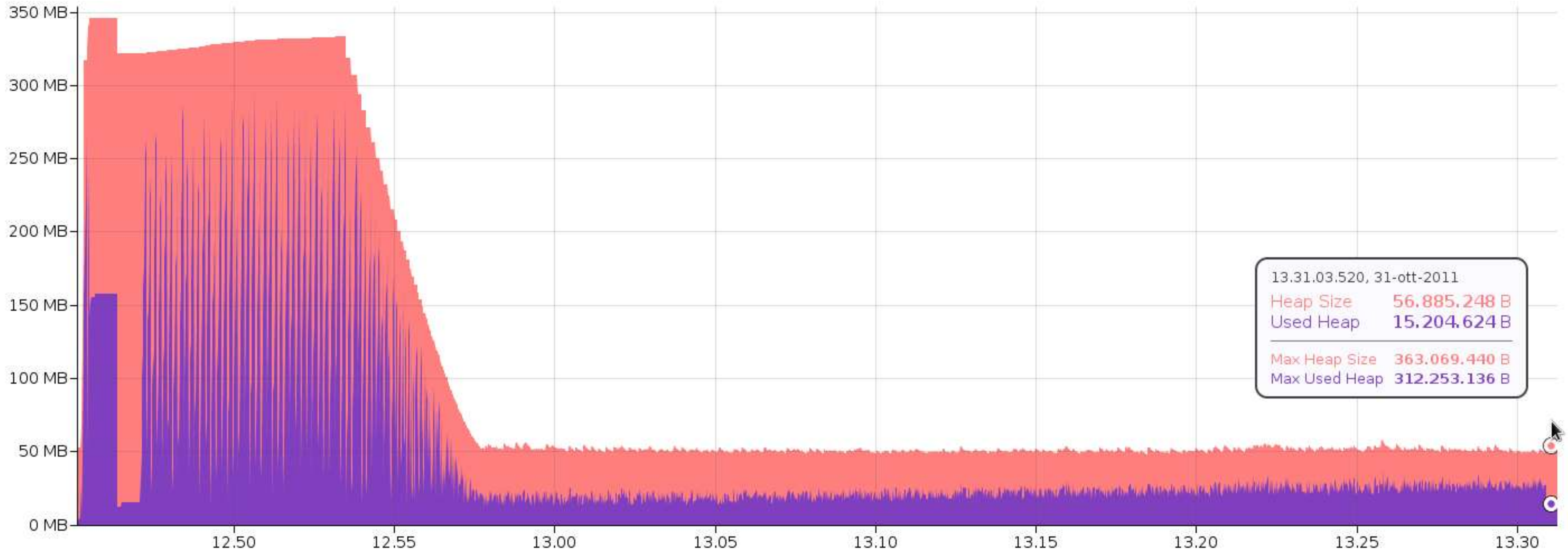


12.27.44.460, 31-ott-2011

Heap Size	98.107.392 B
Used Heap	45.378.136 B
Max Heap Size	386.465.792 B
Max Used Heap	332.371.016 B

apache devicemap

Reduced DeviceMap Memory Load



13.31.03.520, 31-ott-2011

Heap Size	56.885.248 B
Used Heap	15.204.624 B
Max Heap Size	363.069.440 B
Max Used Heap	312.253.136 B

Test Results

- We chose as limited vocabulary the W3C core vocabulary

- The memory load of Apache DeviceMap with limited vocabulary is only about
50MB and after initialization, DeviceMap won't show a memory spike.

- Both APIs identified correctly all the user agents in the HTTP request

SAMPLES

DDR Configuration

DeviceMap DDR implements W3C Simple DDR interfaces. It supports the core vocabulary as specified by the DDR W3C recommendation.

To use DeviceMap Simple DDR you need to configure a property file as follows:

```
oddr.ua.device.builder.path=PATH/BuidlerDataSource.xml
oddr.ua.device.datasource.path=PATH/DeviceDataSource.xml
oddr.ua.device.builder.patch.paths=PATH/BuilderDataSourcePatch.xml
oddr.ua.device.datasource.patch.paths=PATH/DeviceDataSourcePatch.xml
oddr.ua.browser.datasource.path=PATH/BrowserDataSource.xml
ddr.vocabulary.core.path=PATH/coreVocabulary.xml
oddr.vocabulary.path=PATH/oddrVocabulary.xml
oddr.limited.vocabulary.path=PATH/oddrLimitedVocabulary.xml
oddr.vocabulary.device=http://www.openddr.org/oddr-vocabulary
oddr.threshold=70
```

The oddr.threshold property allows the developer to specify the desired confidence to achieve the identification.
In this case we want a confidence of at least 70%.

DDR Service Retrieval

To create the identification service we use the ServiceFactory of W3C DDR-Simple-API

```
initializationProperties.load(context.getResourceAsStream
  ("/WEB-INF/oddr.properties"));
Service identificationService =
ServiceFactory.newService("org.apache.devicemap.ddrsimple.ODDRService",
                          ODDR_VOCABULARY_IRI,
                          initializationProperties);
```

The first argument is the class implementing the DDRService;
the second argument is the default vocabulary used in the identification if the vocabulary is not specified;
the third argument is the SimpleDDR properties file.

New Java Client

While departing from W3C compliance a new Java client offers a bit more flexibility e.g. when it comes to ways of loading device data.

Data can be loaded via:

- URL
- JAR file
- File system

Get client using data from a URL

```
DeviceMapClient client =
  DeviceMapFactory.getClient(LoaderOption.URL,
"http://devicemap-vm.apache.org/data/latest");
```

New Java Client (2)

Get the client using JAR data source

```
DeviceMapClient client =
  DeviceMapFactory.getClient(LoaderOption.JAR);
```

Get client using data from file system

```
DeviceMapClient client =
DeviceMapFactory.getClient(LoaderOption.FOLDER,
"/some/path/devicemap/latest/devicedata");
```

The last option is closest to the W3C DDR client.

DEMO

Outlook

- As of November 2014 after releasing W3C DDR compliant device data and
  new clients beyond W3C for Java, .NET or JavaScript, DeviceMap is ready to graduate from the Apache Incubator.
- You should find it soon under: http://devicemap.apache.org

Conclusion

- Open, free, based on W3C standard. Update to DDR by the community and device manufacturers.
- Identification is not only for the device as a single entity but extended to the web browser
  and the operating system. DeviceMap can identify third parties web browsers and customized operating systems. A developer can specify the vocabulary and the aspects of the property he wants to know.
- Easy migration: Device Repository an some DeviceMap APIs are compatible with W3C DDR

Conclusion (2)

- Based on identification threshold. When a particular browser version is not recognized, DeviceMap returns properties of the nearest version, if this satisfies the defined threshold.
- Developers can choose which properties to be loaded in memory by specifying a limited vocabulary
  → No Waste of Memory.
- Patch Support: if you want to change or update your repository you can do that.
  It is simple to add new properties updating your DDR and your vocabulary.

apache devicemap

Links

- Apache DeviceMap Incubator Site: http://incubator.apache.org/devicemap/
- DeviceMap Sources and Data: http://incubator.apache.org/devicemap/source.html
- Mailing Lists: http://incubator.apache.org/devicemap/mailing-lists.html
- BrowserMap GitHub Mirror: https://github.com/apache/devicemap-browsermap

Links (2)

- Apache Sling DeviceMap Demo on GitHub: https://github.com/raducotescu/devicemap-demo
- Wikimedia LogCapture https://github.com/wikimedia/mediawiki-extensions-DeviceMapLogCapture
- Miri DDR: www.ducis.net/Miri/Ddr

Twitter @wernerkeil
Email wkeil@apache.org