



DSDP Mobile Tools for Java 1.0 Graduation and Release

Eclipse Development Process version 2.3.1 – January 17, 2007

Slide deck v1 – May 29, 2009

Christian Kurzke & Gustavo de Paula

Release Review Agenda



- MTJ Overview and History
- Features
- Non-Code Aspects
- API Status
- Architectural Issues
- Tool Usability
- End-of-life
- Bugzilla
- Standards
- UI Usability
- Schedule
- Communities
- IP Issues
- Next Release Project Plan

A Brief MTJ History



- MTJ Project was submitted by Nokia in 2005
 - Main project sponsors were Nokia and IBM
- Mobile application development environment
 - **CLDC** and **CDC** based devices
 - **Deploy** and **Execute** an application
- Focus was on providing a **framework** that other vendors can extend to create their own JavaME tools
- Release 1.0 was planned to **September 2007**
- MTJ current release is 0.7 from **November 2006**
- Currently renewed discussions about focus of MTJ, new potential contributors, Motorola, RIM
 - MTJ Reboot
 - Evaluate other option to MTJ, such as **EclipseME**

EclipseME – Current Option in Eclipse World



- EclipseME is an Eclipse plug-in for JavaME development
 - Focus on **CLDC/MIDP** (other profiles could be supported)
 - Provides all **basic services** (build, sign, obfuscate, etc.)
 - Provide **some extensibility**, but it is not its main focus
- First public releases on **2003**
- Current version is **1.7.8**
 - Almost **500.000** downloads
- Created and maintained by **Craig Setera**
- Used by all **major** mobile device **manufactures**
 - Nokia, Motorola, Sony Ericsson, etc.
- EclipseME focus is the opposite of original MTJ focus
 - Framework vs. tool

MTJ Reboot Proposed Plan



- Use **EclipseME 1.7.8** as a base for the MTJ development
 - **Less risk**
 - **Large** user community
 - Easier to **create a developer community**
- Re-structure EclipseME code to make it more **flexible**
 - Deployment
 - Signing
 - Packing
- **Port** all interesting features that are already implemented on current **MTJ 0.7** into MTJ 1.0

Features



- [MTJ 1.0 plan available](#)

Main focus is to join Eclipse train and provide a MTJ API

- Besides that the following features were added
 - MIDlet Templates
 - Enhanced MIDlet suite signing solution
 - Fix major MTJ 0.9.X bugs
- MTJ 1.0 provide a refactoring on MTJ project build system
 - Build system was originally from EclipseME
- MTJ is also a core part of [Eclipse Pulsar](#)

Non-Code Aspects



- Requirements and system test cases
 - [Requirements document](#)
 - [Manual System test cases document](#)
 - All documents reviewed with the community
- ISV documentation
 - Includes [Javadoc and a developer documentation](#)
 - Automatically updated from nightly builds
 - [MTJ Webinar](#)
 - [MTJ Video](#)
- Working Example Code
 - All extension points have sample code that shows how to use them
- Conference talks as part of DSDP
 - EclipseCon 2008, ESSE 2008 and EclipseCon 2009

API Status



- MTJ 1.0 defined the first MTJ API release
 - Based on initial EclipseME API with some improvements
 - Add new APIs
- API classes were broken into internal and public classes
 - MTJ 0.9.1
 - Core: 210 classes/interface
 - UI: 339 classes/interface
 - MTJ 1.0
 - Public:
 - Core: 54 classes / interfaces
 - UI: 11 classes / interfaces
 - Internal:
 - Core: 219 classes / interfaces
 - UI: 357 classes / interfaces
- Assumption was to leave as public only a minimum set of classes/interfaces
- All public APIs still declared as provisional

API Status



- Core API is broken into 6 packages
 - Build, Launching, Persistence, Project, SDK and Symbol APIs
- All packages have
 - Full javadoc and examples on how to use the classes
 - Only manual test cases (there was not time to implement automated test cases)
 - Each package have at least two clients inside MTJ
 - Commercial clients are under development on Nokia, RIM and Motorola

API Status



- 8 extension points
 - 2 originally from EclipseME
 - 6 added on MTJ
- All extensions have
 - Full detailed description
 - A sample plugin that shows how to use it
 - At least one client inside MTJ
 - Commercial clients are under development on Nokia, RIM and Motorola

Architectural Issues



- MTJ UI still need to be refactored
 - Most of the work done until focus on MTJ core
 - Still need to do some cleaning on MTJ UI
 - Remove unnecessary classes
 - Refactore it
 - Provide more extensions
- Need to provide more than MIDP support
 - MTJ API enables non-MIDP development environment
 - It is necessary to add at least one more on a next release
- Need Unit Tests
 - Original EclipseME did not have a lot of unit tests
 - This effort depends on creating a well defined API

Tool Usability



- As a tool MTJ inherited all the good usability that was already available on EclipseME
- Developer is able to execute all main operations that are associated with JavaME development
 - Create MIDlet suites and MIDlets
 - Edit JAD file
 - Build & Sign the MIDlet suite
 - Import / Edit JavaME SDKs
 - Run & Debug MIDlet
- Several improvements were made on the workflow to make it easy to execute some of those tasks

End-of-life



- There was no well defined API on MTJ 0.9.1
- The extension points available on MTJ 0.9.1 are still available on MTJ 1.0

Bugzilla



- 129 bugs resolved up to 13-MAY-2009
- 5 main bugs categories
 - Galileo → all Galileo Must-dos were tracked on MTJ bugzilla
 - API → all bugs related to API definition were tracked with this category
 - Fix and Improve → fix issues from other releases and do some UI improvements;
 - MIDlet Templates → MIDlet templates new feature
 - Signing Enhancements → signing enhancements new feature
- Extensive and open discussions on bugzilla
- No major bug on final 1.0 release

Standards



- Mobile Information Device Profile Specification
 - Version 2.1: <http://www.jcp.org/en/jsr/detail?id=118>
- Connected Limited Device Configuration Specification
 - Version 1.1: <http://www.jcp.org/en/jsr/detail?id=139>
- UEI Specification is implemented on MTJ
 - Version 1.0.2 http://java.sun.com/j2me/docs/uei_specs.pdf

UI Usability



- Accessibility were not the focus of this release
 - This is on the scope of MTJ 1.0
- All visible UI strings are externalized
 - Externalization done via Eclipse NLS
- MTJ is also part of Eclipse Babel

Schedule



- Original project plan posted on 20-Jan-2009
- Revised with MTJ community and the following weeks
 - All scope on original plan was accomplished, except from
 - Unit Tests
 - GUI Editor
 - Effort to define API was under-estimated, this impacted the Unit test development
 - Sybase wasn't able to commit to GUI Editor
- All milestone dates were hit
- Focus on API related issues and major bug fixes

Communities



- Contributors
 - Initial code base from EclipseME
 - Current Major code contribution from Motorola, Sybase and Nokia
 - 5 active committers: 3 from Motorola 1 from Nokia and 1 individual
 - Mailing list participation from Motorola, Sybase, Nokia, RIM and other individuals
- Adopters
 - MTJ 0.9.1 is part of MOTODEV Studio for JavaME v2.2
 - Other vendors are working on their extensions
- Users
 - MTJ 0.9 had approximately 5500 downloads (from october/2008 until beginning on May/2009)
 - MTJ 0.9.1 had approximately 21000 downloads (from december/2008 until beginning on May/2009)

IP Issues



As per the Eclipse IP Policy, the project verifies that:

- ... the about files and use licenses are in place as per the Guidelines
- ... all contributions (code, documentation, images, etc) have been committed by individuals who are Members of the Foundation and are abiding by the Eclipse IP Policy (training through Committer HOWTO)
- ... all significant contributions have been reviewed by the Foundation's legal staff – even if written by committers prior to joining Eclipse
- ... third-party libraries, have been documented in the release and reviewed by the Foundation's legal staff
- ... all contribution questionnaires have been completed
- ... the "provider" field of each plug-in is set to "Eclipse.org - DSDP"
- ... the "copyright" field of each plug-in is set to the copyright owner
- ...there are no 3rd party logos or fonts to be licensed under the EPL
- See the IP Log at
<http://www.eclipse.org/dsdp/mtj/development/mtj-log.csv>
<http://www.eclipse.org/dsdp/mtj/ipLog.php>

Future Plans



- MTJ needs to establish itself as the Eclipse mobile IDE environment
 - Add other mobile platforms support (CLDC, eRCP, Android, etc.)
- MTJ Team needs to work with the community to show the features that are currently available on MTJ 1.0
 - Conferences
 - Articles
 - Tutorials



Thank You