DSDP Target Management 3.1 Release Review Minideck

Eclipse Development Process version 2.4 – August 20, 2008
Slide deck v1 – June 3, 2009

Martin Oberhuber, Wind River
TM Project Lead

DSDP-TM 3.1 Features

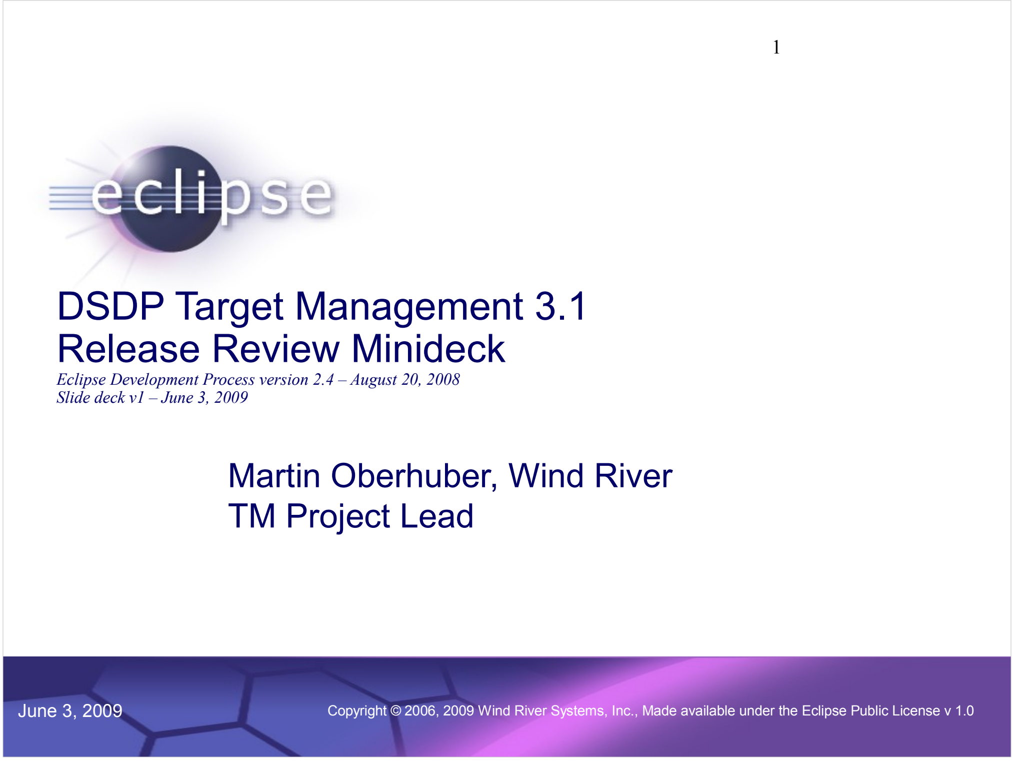
- TM 3.1 New Features
  - Platform/Team Synchronize integration (GSoC contribution)
  - Generic Terminal now also for telnet
  - Several smaller performance / usability improvements in RSE
  - Most feature work and community interest in TCF (still incubating)
- API Quality:
  - Few well-reviewed API additions backed by API Tooling.
    - W/o samples & tests: 833 API types / 1354 non-API (3.0: 935 API / 1476 non-API)
  - Fully binary compatible with TM 3.0
- TM 3.1 project size
  - RSE: 356 kLOC + TCF: 162 kLOC (R3.0: 346k + 97k)

DSDP-TM 3.1 EOL and Community

- Non-Code Aspects
  - Full Documentation, Tutorials, FAQs, Example Code, …
  - Detailed build notes with migration notes on each milestone
- End-of-Life issues:
  - RemoteCDT moving into CDT (but still very active)
  - TM Discovery no longer maintained
  - Some parts of RSE client moving from Java 1.4 to Java 5 (server still 1.4)
- Community and Committer Diversity:
  - 10 committers (5 WindRiver, 4 IBM, 1 Montavista) – was 11 in 3.0
  - 18 additional contributors, mostly IBM – was 23 in 3.0
  - Well known and respected in the Community, part of JEE package
- Bugzilla
  - 3.1 stream: 224 issues fixed / 690 open (3.0: 441 fixed / 671 open)

DSDP-TM 3.1 Process and Architecture

- Process
  - Full process docs on the Web; adopting Modeling build for Releng
- IP Clearance and Licenses:
  - All licenses and about files are in place as per the Eclipse Development Process, the Due Diligence Process was followed for all contributions.
- Architectural Issues
  - Legacy code still not fully cleaned up – much Platform “internal” access
    - Need more UI / Non-UI separation for headless and RCP usage
    - Need more Unit Tests (hard for UI-heavy parts)
  - Overlaps with other projects - Many remote access APIs
    - E.g. Remote File Service – 5 APIs: Platform EFS, ECF fileshare, TPTP Agent File Interfaces, TCF, RSE IFileService
    - Talking with all those projects; absorbing / bridging
  - “Remote Development (RDT)” effort is disconnected at IBM / PTP
- Future:
  - Likely shooting for TM 3.2 next year – Focus on Multicore, TCF



DSDP Target Management 3.1 Release Review Minideck

Eclipse Development Process version 2.4 – August 20, 2008
Slide deck v1 – June 3, 2009

Martin Oberhuber, Wind River
TM Project Lead

DSDP-TM 3.1 Features

- TM 3.1 New Features
  - Platform/Team Synchronize integration (GSoC contribution)
  - Generic Terminal now also for telnet
  - Several smaller performance / usability improvements in RSE
  - Most feature work and community interest in TCF (still incubating)
- API Quality:
  - Few well-reviewed API additions backed by API Tooling.
    - W/o samples & tests: 833 API types / 1354 non-API (3.0: 935 API / 1476 non-API)
  - Fully binary compatible with TM 3.0
- TM 3.1 project size
  - RSE: 356 kLOC + TCF: 162 kLOC (R3.0: 346k + 97k)

DSDP-TM 3.1 EOL and Community

- Non-Code Aspects
  - Full Documentation, Tutorials, FAQs, Example Code, …
  - Detailed build notes with migration notes on each milestone
- End-of-Life issues:
  - RemoteCDT moving into CDT (but still very active)
  - TM Discovery no longer maintained
  - Some parts of RSE client moving from Java 1.4 to Java 5 (server still 1.4)
- Community and Committer Diversity:
  - 10 committers (5 WindRiver, 4 IBM, 1 Montavista) – was 11 in 3.0
  - 18 additional contributors, mostly IBM – was 23 in 3.0
  - Well known and respected in the Community, part of JEE package
- Bugzilla
  - 3.1 stream: 224 issues fixed / 690 open (3.0: 441 fixed / 671 open)

DSDP-TM 3.1 Process and Architecture

- Process
  - Full process docs on the Web; adopting Modeling build for Releng
- IP Clearance and Licenses:
  - All licenses and about files are in place as per the Eclipse Development Process, the Due Diligence Process was followed for all contributions.
- Architectural Issues
  - Legacy code still not fully cleaned up – much Platform "internal" access
    - Need more UI / Non-UI separation for headless and RCP usage
    - Need more Unit Tests (hard for UI-heavy parts)
  - Overlaps with other projects - Many remote access APIs
    - E.g. Remote File Service – 5 APIs: Platform EFS, ECF fileshare, TPTP Agent File Interfaces, TCF, RSE IFileService
    - Talking with all those projects; absorbing / bridging
  - "Remote Development (RDT)" effort is disconnected at IBM / PTP
- Future:
  - Likely shooting for TM 3.2 next year – Focus on Multicore, TCF