Device Software Development Platform (DSDP) Project

Doug Gaff
Wind River Systems
23 January 2007

Agenda

- DSDP Overview
- General embedded support
  - Device Debugging (DD)
  - Target Management (TM)
- Mobile Java
  - Mobile Tools for the Java Platform (MTJ)
  - Embedded Rich Client Platform (eRCP)
- Mobile C/C++
  - Native Application Builder (NAB)
  - Tools for Mobile Linux (TmL)
- Getting Involved

DSDP Overview

www.eclipse.org/dsdp

- Device Software is ***software than runs on an embedded operating system inside a larger physical product.***
- DSDP Mission:

  Create an open, extensible, scalable, and standards-based development platform to address the needs of the device (embedded) software market by enabling developers and vendors to create differentiated, specialized, and interoperable solutions to help customers and users of Eclipse-based products develop device software faster, better, and at lower cost.
- DSDP intends to address development personas
  - Hardware Bring-up
  - Platform Software Development
  - Target-based Application Software Development
- DSDP builds on existing Eclipse technology: Eclipse Platform, CDT, JDT, etc.

DSDP History

EclipseCon 2005	Device software tools vendors discuss need for more embedded-specific functionality in Eclipse.
Mar 2005	Wind River proposes DSDP.
Jun 2005	Eclipse Board votes to create the DSDP project. Two sub-projects created: Device Debugging (DD) and Target Management (TM).
Jan 2006	Two additional sub-projects created: Mobile Tools for the Java Platform (MTJ) and Native Application Builder (NAB).
July 2006	Embedded Rich Client Platform (eRCP) moves from Technology to DSDP
Aug 2006	Tools for Mobile Linux (TmL) project proposed
Sept 2006	eRCP 1.0 released
Oct 2006	TM 1.0, MTJ 0.7, NAB 0.96 released
Dec 2006	TmL passes creation review, TM 1.0.1
Jan 2006	eRCP 1.0.1

DSDP Leadership

Doug Gaff
PMC Lead
DD Lead

WIND RIVER

Mika Hoikkala
MTJ Lead

NOKIA
Connecting People

Christian Kurzke
TmL Lead

Shigeki Moride
NAB Lead

FUJITSU

Martin
Oberhuber
TM Lead

WIND RIVER

Mark Rogalski
eRCP Lead

IBM

eclipse PluginFest

DSDP Stats

- 6 Projects – DD, eRCP, MTJ, NAB, TM, TmL
- Over 550k Physical Lines of Code (not incl. comments)
- Over 40 committers representing (in alphabetical order):

Accelerated Technology A Mentor Graphics Division · freescale semiconductor · FUJITSU · IBM · Motorola

NOKIA Connecting People · palmsource · symbian · Texas Instruments · WIND RIVER

- Other companies
  - Curtiss-Wright, Intel, QNX, ARM, AMI Semiconductor, MontaVista, SonyEricsson, Sybase, ShareME Technologies, and others.
- Open source projects
  - EclipseME and Antenna
- Press coverage
  - Embedded Technology Journal, SDTimes, EclipseSource, DSO.com, LinuxDevices.com, EETimes, Embedded.com, and more

Agenda

- DSDP Overview
- General embedded support
  - Device Debugging (DD)
  - Target Management (TM)
- Mobile Java
  - Mobile Tools for the Java Platform (MTJ)
  - Embedded Rich Client Platform (eRCP)
- Mobile C/C++
  - Native Application Builder (NAB)
  - Tools for Mobile Linux (TmL)
- Getting Involved

Device Debugging (DD)

www.eclipse.org/dsdp/dd

- Mission: ***Build enhanced debug models, API's, and views that augment the Eclipse Debug Platform in order to address the added complexities of device software debugging.***

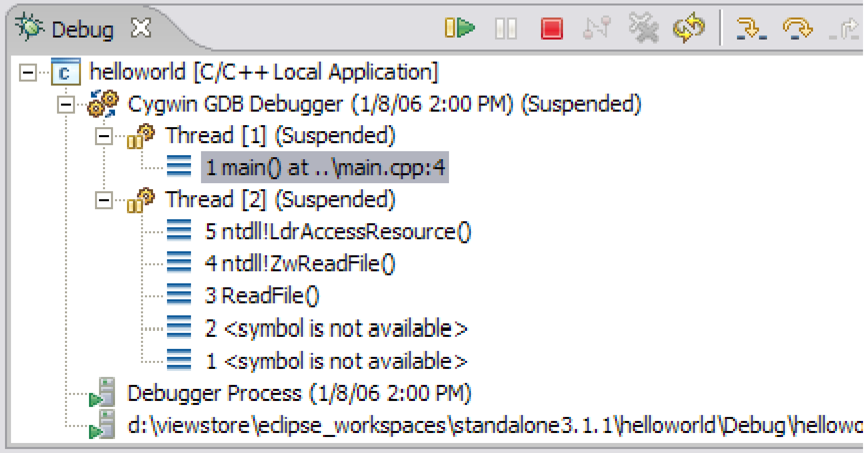
- Wind River (lead), IBM, Mentor Graphics, Nokia, PalmSource, Symbian, TI, QNX, Freescale

- Tasks
  - ☑ Modify the Eclipse Debug Model Interfaces for customized embedded debugger implementations. (Released in Eclipse 3.2 as provisional API's.)
  - ☑ Build requirements and use cases for device software development needs in Eclipse.
  - ☑ Enhance the platform memory view.
  - Provide a new Debug Model implementation that takes a more modular approach to connecting debugger backends into Eclipse. This is called Debugger Services Framework (DSF).
  - Enhance the debugger views for multi-core and multi-process support and provide specific improvements in those views for embedded development.
  - Integrate with the SPIRIT consortium for tooling and debugger data file specification.
  - *Provide the next generation implementation for CDT's MI debugger – future.*

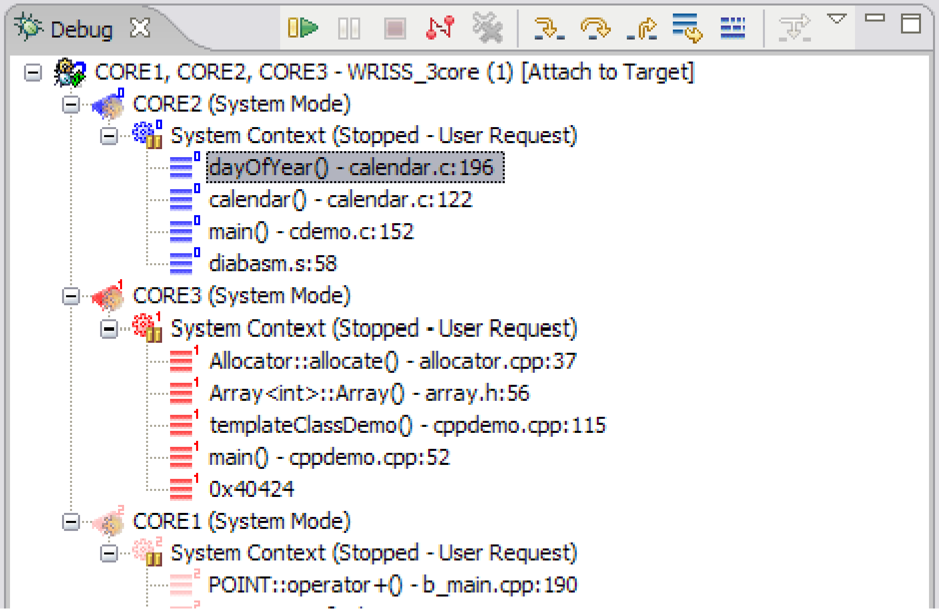
DD – more detail

- The new Eclipse 3.2 Debug Model
  - A flexible debug element hierarchy
  - Model driven view updates
  - Asynchronous interactions between UI and debug model
  - Flexible view wiring (e.g. input to variables view)
  - The ability to debug multiple sessions simultaneously
- The Debugger Services Framework (DSF)
  - Concurrency – ensures thread-safety and fast responsiveness for slow debugger operations like stepping and debugger view population
  - Services – provides plugability of individual debugger components like register, memory, breakpoints,etc.
  - Data Model – for retrieving data and populating views.
- Release Plans
  - Europa train milestones starting with M4
  - 0.9 release – June 07 on Europa train

DD: Eclipse 3.1 vs. Eclipse 3.2+ Debug Model



```
helloworld [C/C++ Local Application]
  Cygwin GDB Debugger (1/8/06 2:00 PM) (Suspended)
    Thread [1] (Suspended)
      1 main() at ..\main.cpp:4
    Thread [2] (Suspended)
      5 ntdll!LdrAccessResource()
      4 ntdll!ZwReadFile()
      3 ReadFile()
      2 <symbol is not available>
      1 <symbol is not available>
  Debugger Process (1/8/06 2:00 PM)
  d:\viewstore\eclipse_workspaces\standalone3.1.1\helloworld\Debug\hellowo
```



```
CORE1, CORE2, CORE3 - WRISS_3core (1) [Attach to Target]
  CORE2 (System Mode)
    System Context (Stopped - User Request)
      dayOfYear() - calendar.c:196
      calendar() - calendar.c:122
      main() - cdemo.c:152
      diabasm.s:58
  CORE3 (System Mode)
    System Context (Stopped - User Request)
      Allocator::allocate() - allocator.cpp:37
      Array<int>::Array() - array.h:56
      templateClassDemo() - cppdemo.cpp:115
      main() - cppdemo.cpp:52
      0x40424
  CORE1 (System Mode)
    System Context (Stopped - User Request)
      POINT::operator+() - b_main.cpp:190
```

Eclipse 3.1

- Rigid debug element hierarchy (Target – Process – Thread – Stack Frame)
- Fixed view update policies
- Fixed debugger actions

➡

Eclipse 3.2 and beyond

- Customizable debug hierarchy
- Model-driven view content and update policies
- Retargettable debugger actions

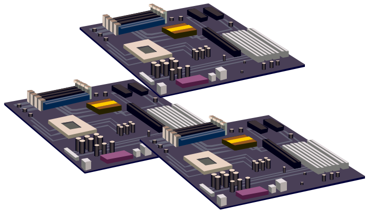
Target Management (TM)

www.eclipse.org/dsdp/tm

- Mission: ***Create data models and frameworks to configure and manage embedded systems, their connections, and their services.***
- Wind River (lead), IBM, MontaVista, PalmSource, Symbian
- LANL, Freescale, Mentor Graphics, Nokia, TI, QNX

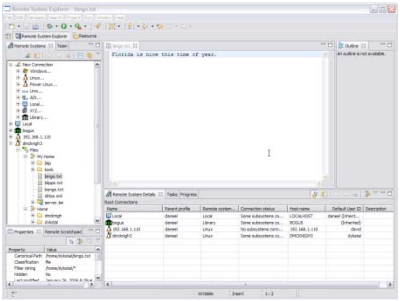
Remote Computer Systems…

- *Targets (Locally connected, shared, fielded)*
- *Hosts (Grids, farms, nodes)*



… and developing software on them

- *Build, connect, get status*
- *Download, run, debug, test*
- *Upload*



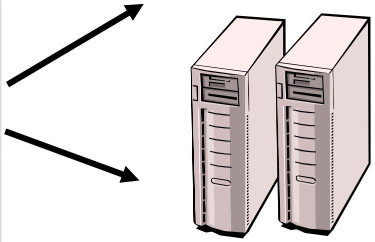
```
Florida is nice this time of year.
```

An outline is not available.

Root Connections

Name	Parent profile	Remote system…	Connection status	Host name	Default User ID	Description
Local	daneel	Local	Some subsystems co…	LOCALHOST	daneel (Inherit…	
bogus	daneel	Library	Some subsystems co…	BOGUS	(Inherited)	
192.168.1.110	daneel	Linux	No subsystems conn…	192.168.1.110	david	
dmcknigh3	daneel	Linux	Some subsystems co…	DMCKNIGH3	dykstal	

Property	Value
Canonical Path	/home/dykstal/bingo.txt
Classification	file
Filter string	/home/dykstal/*
Hidden	No
Last modified	January 26, 2006 8:39:x

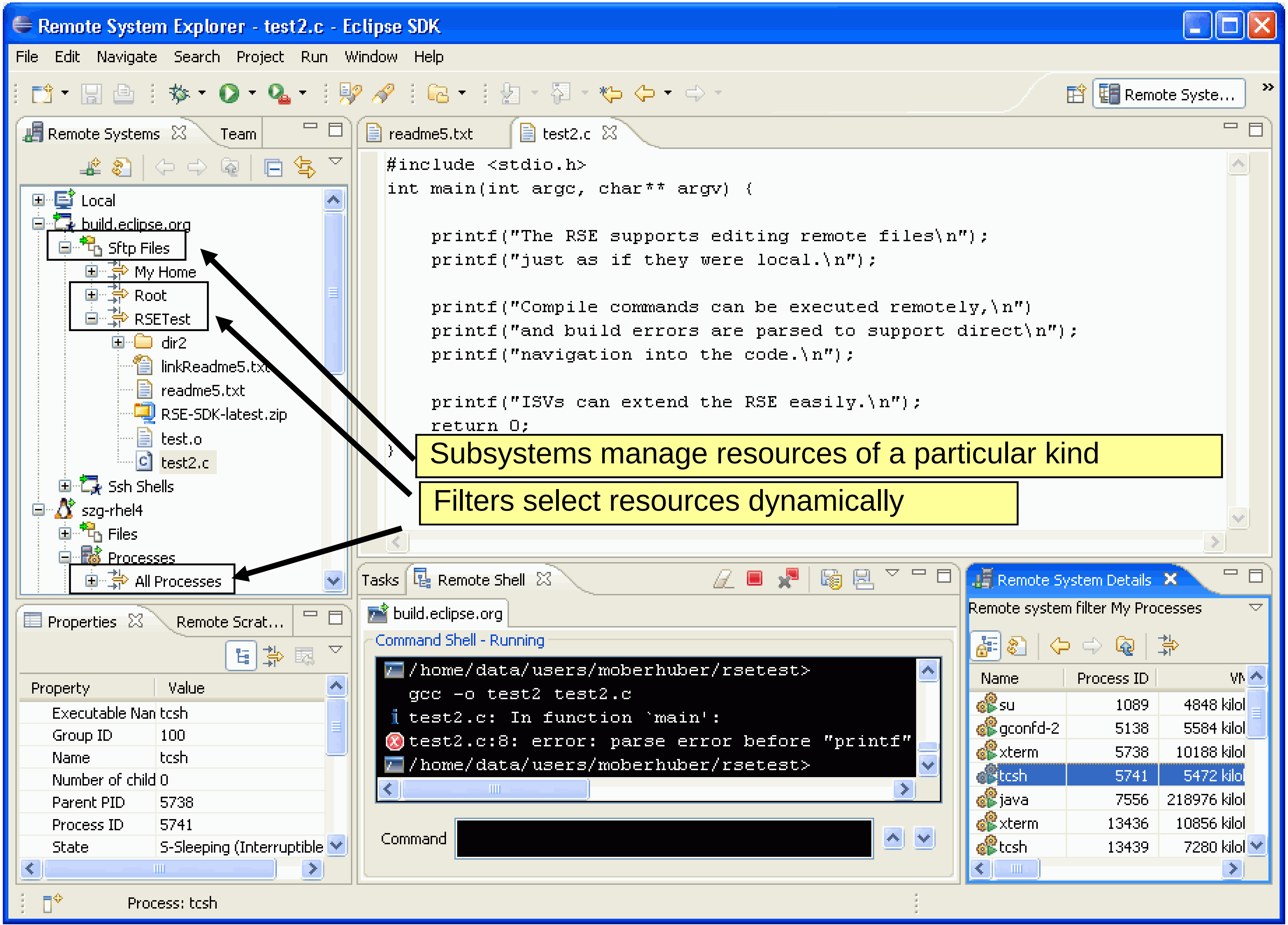


TM Features

- Features in 1.0.1 (December 06)
  - IBM RSE Framework
  - Dstore, FTP, ssh connection types
  - Integrate Jakarta Commons Net library for FTP access
  - Provide complete user and ISV documentation, tutorials and examples
  - CDT remote launch capabilities
  - Ansi terminal view & serial connection
  - Test on Windows, Linux, Solaris, Mac
  - Zeroconf Discovery, EFS Integration – Preview Release

TM 2.0 Plan – June 07 on Europa train

- Committed
  - Contribute user actions, compile commands, and import/export from RSE7
  - Allow encoding of remote files to be specified
  - Integrate the TM Terminal View with RSE
  - Improve Discovery and Autodetect in RSE
- Proposed
  - Adopt Eclipse Platform 3.3 concepts in RSE
  - Fix and improve the RSE EFS integration
  - Improve RSE SystemType and New Connection Wizard flexibility
  - Optimize APIs - Remove obsolete API
  - Improve UI/Non-UI splitting in RSE. Support headless launches
  - Improve the Remote File Service APIs
  - RSE should be more service-oriented
  - Improve the RSE default Persistence Provider
  - Add full support for Macintosh



```
#include <stdio.h>
int main(int argc, char** argv) {

    printf("The RSE supports editing remote files\n");
    printf("just as if they were local.\n");

    printf("Compile commands can be executed remotely,\n")
    printf("and build errors are parsed to support direct\n");
    printf("navigation into the code.\n");

    printf("ISVs can extend the RSE easily.\n");
    return 0;
}
```

```
/home/data/users/moberhuber/rsetest>
gcc -o test2 test2.c
test2.c: In function `main':
test2.c:8: error: parse error before "printf"
/home/data/users/moberhuber/rsetest>
```

Property	Value
Executable Nan	tcsh
Group ID	100
Name	tcsh
Number of child	0
Parent PID	5738
Process ID	5741
State	S-Sleeping (Interruptible

Name	Process ID	VM
su	1089	4848 kilol
gconfd-2	5138	5584 kilol
xterm	5738	10188 kilol
tcsh	5741	5472 kilol
java	7556	218976 kilol
xterm	13436	10856 kilol
tcsh	13439	7280 kilol

Subsystems manage resources of a particular kind

Filters select resources dynamically

Agenda

- DSDP Overview
- General embedded support
  - Device Debugging (DD)
  - Target Management (TM)
- Mobile Java
  - Mobile Tools for the Java Platform (MTJ)
  - Embedded Rich Client Platform (eRCP)
- Mobile C/C++
  - Native Application Builder (NAB)
  - Tools for Mobile Linux (TmL)
- Getting Involved

Mobile Tools for the Java Platform (MTJ)

www.eclipse.org/dsdp/mtj

- Mission: *Extend existing Eclipse frameworks to support mobile device Java application development, including a* ***device and emulator framework****, a* ***deployment framework****,* ***generic build processes*** *for mobile application development,* ***mobile device debugging****, application creation wizards, UI design tools, localization, optimization, and security.*

- Major participants
  - Nokia (lead), IBM, SonyEricsson, EclipseME project

- Other participants
  - Sybase, Apogee Software, Sprint, Sysline Inc, Antenna, ShareME Technologies

- Release plans
  - 0.7 release in October 2006
  - 1.0 project plan in September 2007 (tentative)

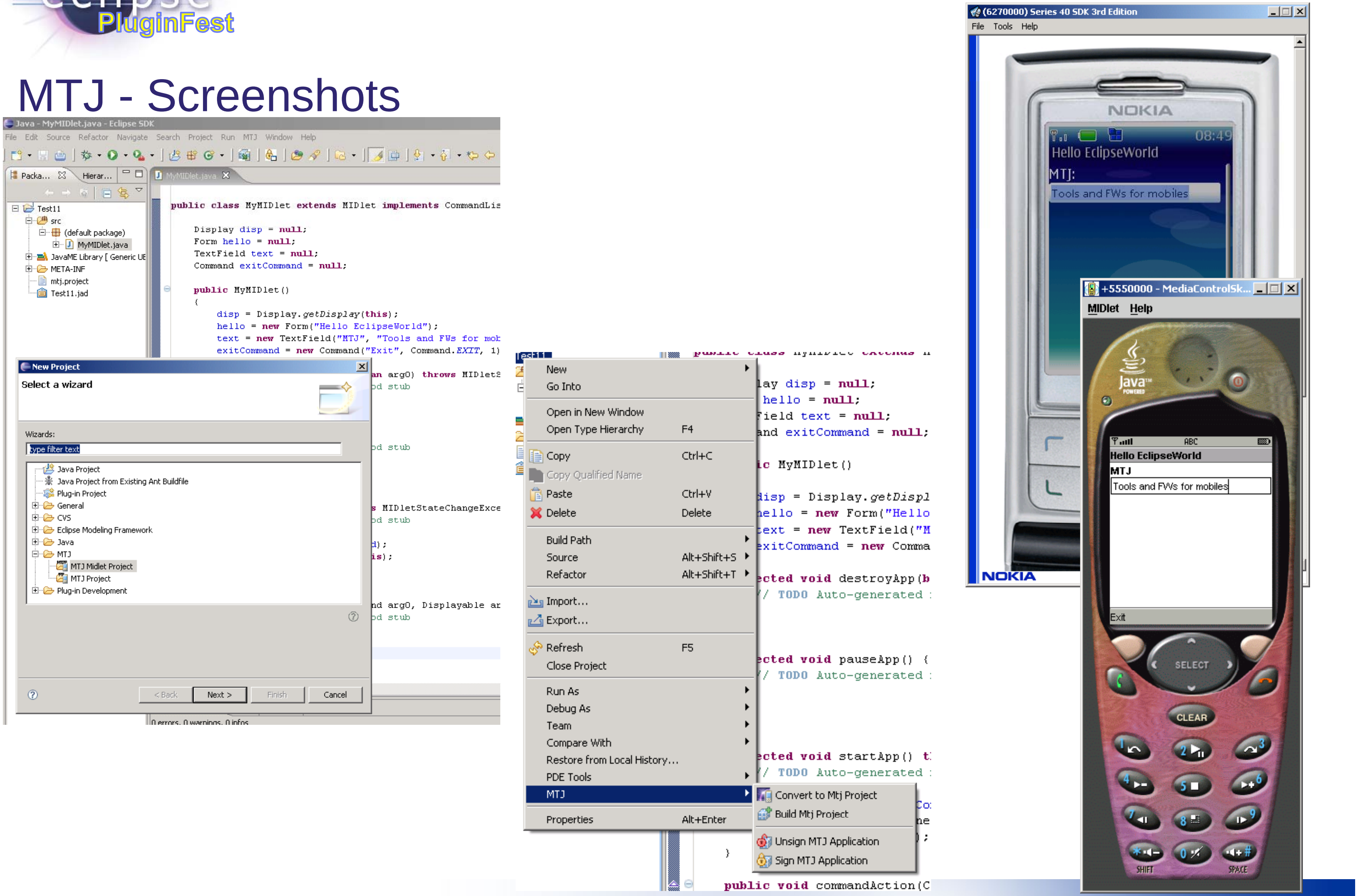
MTJ 0.7 Features (Oct 2006)

- To create Eclipse Mobile Java Tools platform that vendors can extend to support their devices. Extensibility currently includes
  - Runtime management framework
    - adding device adapter to manage emulators + real devices
  - Build framework, customized and extensible build process
    - Packaging (CDC, CLDC, Java in Palm devices, Java in Nokia devices,…)
    - Signing (differences between devices)
  - Deployment framework
- Provide default tools to develop mobile Java applications.
  - Create a project
  - Create a code
  - Compile
  - Package
  - Run in emulator
  - Signing
  - Transfer to real Device (only Nokia)
- Provide User and developer documentation

MTJ Future Plans (proposed)

- Version 1.0
  - With needed quality
  - With needed features (to have “full” development environment)
  - With needed documentation
  - Schedule driven
- Fixes and Enhancements to R1
  - Finalize the APIs
  - Enhance documentation
  - Project based preferences
  - Support for non UEI (unified emulator interface) SDKs (user provides the information)
  - Enhance build mechanism to manage resources
  - Bug fixes
- Visual Designers
  - LCDUI
  - eSWT (eRCP now part of the DSDP)
- Fragmentation
  - Build time solutions e.g. pre-processing
  - Device Information database access
- Obfuscation
  - External obfuscators are possible to plug-in
  - Possible one default integrated
- New Profiles and Configurations
  - CDC
  - Foundation Profile, Personal Profile (no AWT visual editor), Personal Basis Profile
  - MIDP 2.1 support, MIDP 3.0 support (not included, but may come to the picture)
- Mobile JUnit
- Localization (low)
  - Manage resources
  - Visual designer support

MTJ - Screenshots



Java - MyMIDlet.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run MTJ Window Help
Packa... Hierar...
Test11
src
(default package)
MyMIDlet.java
JavaME Library [Generic UE
META-INF
mtj.project
Test11.jad
MyMIDlet.java

```
public class MyMIDlet extends MIDlet implements CommandLis

    Display disp = null;
    Form hello = null;
    TextField text = null;
    Command exitCommand = null;

    public MyMIDlet()
    {
        disp = Display.getDisplay(this);
        hello = new Form("Hello EclipseWorld");
        text = new TextField("MTJ", "Tools and FWs for mob
        exitCommand = new Command("Exit", Command.EXIT, 1)
```

```
lay disp = null;
hello = null;
Field text = null;
and exitCommand = null;

ic MyMIDlet()

disp = Display.getDispl
hello = new Form("Hello
text = new TextField("M
exitCommand = new Comma

ected void destroyApp(b
// TODO Auto-generated

ected void pauseApp() {
// TODO Auto-generated

ected void startApp() t
// TODO Auto-generated

public void commandAction(C
```

Embedded Rich Client Platform

www.eclipse.org/ercp

- Mission: ***Extend Eclipse's Rich Client Platform to embedded and mobile devices.***
- IBM (lead), Nokia, Motorola
- Features
  - OSGI, eSWT + mobile extensions, eJFace, eWorkbench, eUpdate, microXML.
  - Utilizes RCP application framework model
  - Reduces RCP size/function to fit on devices
  - Pushes changes back to core components to enable running those components on JME CDC/Foundation Profile
  - Adds components to enable application binary compatibility across a range of devices with different input mechanisms and screen types/sizes

eRCP Benefits

The next step up in Java platforms for devices

- Extensive rich UI capabilities
- Higher level of device abstraction
- Integration with native platform look and feel
- Brings OSGi service oriented features to devices
  - Dynamic install/uninstall
  - Sharing of services
- Puts the Eclipse programming model on devices – developers can use their existing knowledge and skills

eRCP Platforms

- Release 1.0.1 (Jan 2007)
  - Windows™ Desktop
  - Windows Mobile 2003/5
  - Nokia Series 80 platform
- Next Release (in Europa time frame)
  - Nokia S60 platform (coming very soon)
  - Linux Qte (in progress)
  - GTK, UIQ, … under consideration

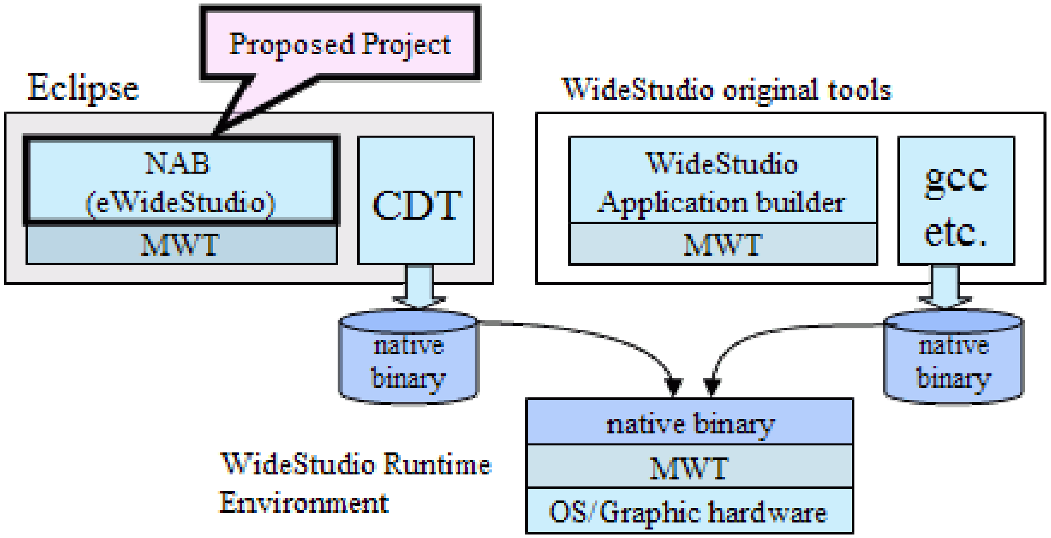
Agenda

- DSDP Overview
- General embedded support
  - Device Debugging (DD)
  - Target Management (TM)
- Mobile Java
  - Mobile Tools for the Java Platform (MTJ)
  - Embedded Rich Client Platform (eRCP)
- Mobile C/C++
  - Native Application Builder (NAB)
  - Tools for Mobile Linux (TmL)
- Getting Involved

Native Application Builder (NAB) www.eclipse.org/dsdp/nab

- Mission: ***Create a C++ GUI builder for embedded operating systems, similar to eSWT for Java.***
- Fujitsu (lead), WideStudio team, Eclipse Japan Working Group
- Born out of the WideStudio/MWT open source project (www.widestudio.org).
  - Thousands of mailing list subscribers from 20 countries
  - > 800,000 downloads
- WideStudio is a GUI application builder for multiple host and embedded operating systems.
- MWT (Multiplatform Widget Toolkit) is the run-time library that implements the GUI objects. Available at www.widestudio.org.
  - X11, Windows, Linux, MacOS, FreeBSD
  - WinCE
  - ITRON, BTRON, T-Engine
- Release plans
  - 0.9.6 available now
  - 1.0 project plan in progress (June 07, but not in Europa)

NAB: Architecture

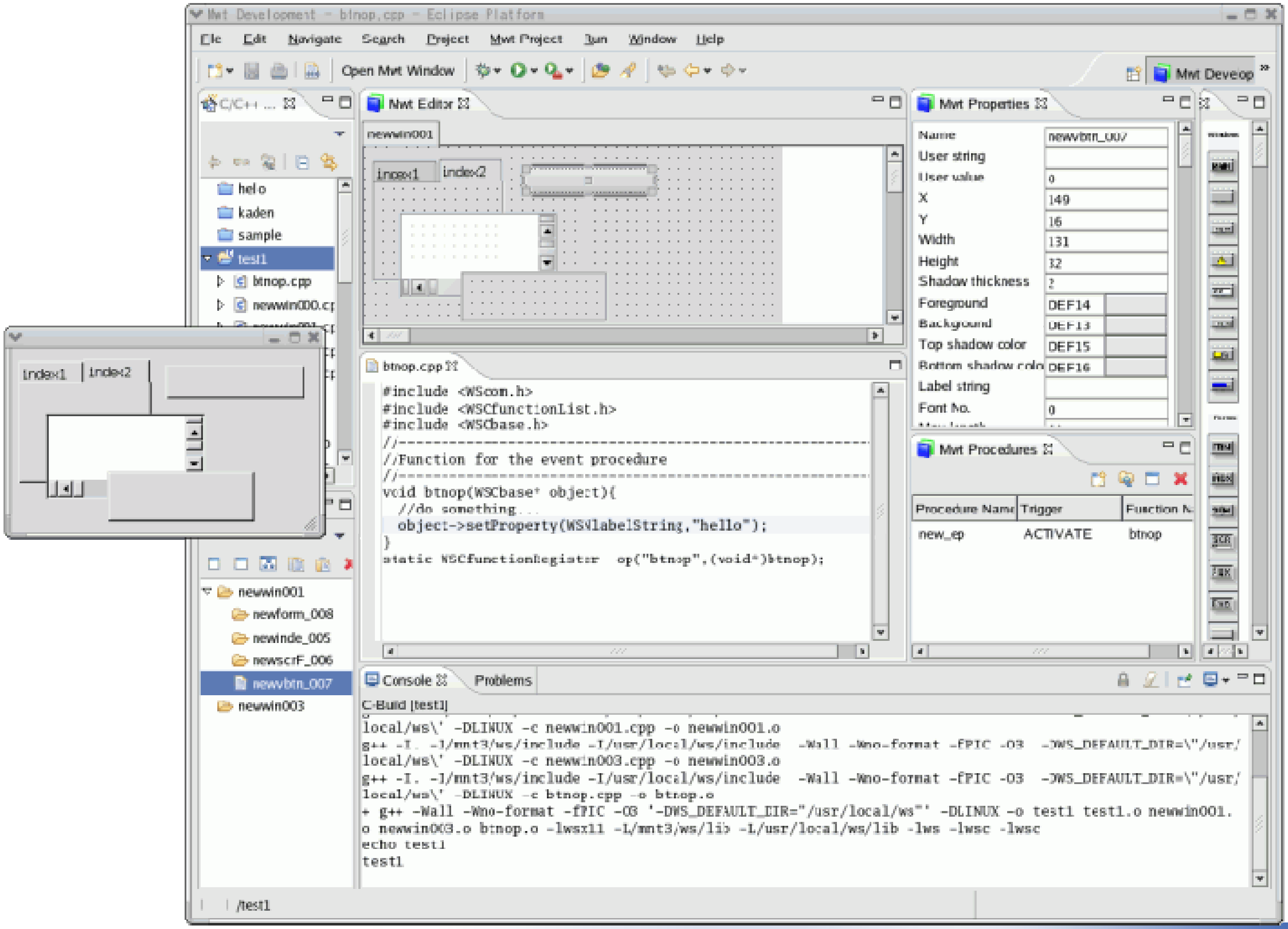


NAB Provides

- GUI editing
- C++ source code generation
- Application build and debug with CDT

Users pick the desired deployment environment and download the appropriate MWT runtime code from widestudio.org.

NAB: Visual Editor



btnop.cpp

```
#include <WScom.h>
#include <WSCfunctionList.h>
#include <WSCbase.h>
//-----------------------------------------------------------------
//Function for the event procedure
//-----------------------------------------------------------------
void btnop(WSCbase* object){
  //do something...
  object->setProperty(WSNlabelString,"hello");
}
static WSCfunctionRegister  op("btnop",(void*)btnop);
```

Mwt Properties

Name	newvbtn_007
User string	
User value	0
X	149
Y	16
Width	131
Height	32
Shadow thickness	2
Foreground	DEF14
Background	DEF13
Top shadow color	DEF15
Bottom shadow colo	DEF16
Label string	
Font No.	0

Mwt Procedures

Procedure Name	Trigger	Function N
new_ep	ACTIVATE	btnop

Console | Problems

C-Build [test1]

```
local/ws\' -DLINUX -c newwin001.cpp -o newwin001.o
g++ -I. -I/mnt3/ws/include -I/usr/local/ws/include  -Wall -Wno-format -fPIC -O3  -DWS_DEFAULT_DIR=\"/usr/
local/ws\' -DLINUX -c newwin003.cpp -o newwin003.o
g++ -I. -I/mnt3/ws/include -I/usr/local/ws/include  -Wall -Wno-format -fPIC -O3  -DWS_DEFAULT_DIR=\"/usr/
local/ws\' -DLINUX -c btnop.cpp -o btnop.o
+ g++ -Wall -Wno-format -fPIC -O3 '-DWS_DEFAULT_DIR="/usr/local/ws"' -DLINUX -o test1 test1.o newwin001.
o newwin003.o btnop.o -lwsx11 -L/mnt3/ws/lib -L/usr/local/ws/lib -lws -lwsc -lwsc
echo test1
test1
```

Tools for Mobile Linux (TmL) www.eclipse.org/proposals/tml/

- Motorola (lead)
- Creation frameworks and tools for entire life-cycle C/C++ application development targeted at mobile Linux platforms.
  - Design
    - Focus on modeling
  - Development
    - Cross-compilation of OS, middleware, and applications
    - Focus on mobile device services
  - Debug
    - Cross debugging
    - Device emulation support
  - Deployment
    - Application testing
    - Code Signing

TmL continued

- Will reuse and extend existing technology
  - Modeling
  - CDT
  - DD, TM, MTJ
  - TPTP
- Initial plans
  - Passed creation review in December 2006
  - Working on provisioning now
  - First technology: Mobile Linux Emulator Framework
  - Gathering community and building initial development team

Agenda

- DSDP Overview
- General embedded support
  - Device Debugging (DD)
  - Target Management (TM)
- Mobile Java
  - Mobile Tools for the Java Platform (MTJ)
  - Embedded Rich Client Platform (eRCP)
- Mobile C/C++
  - Native Application Builder (NAB)
  - Tools for Mobile Linux (TmL)
- Getting Involved

Getting Involved

- Start here: www.eclipse.org/dsdp
- Helping existing projects
  - As with all Eclipse projects, there’s a lot of work left to do.
  - Contributors are needed to help with the open-source implementations.
  - Users are needed to verify that commercial products can be built on top of the frameworks.
- DSDP new project ideas
  - Automotive tooling
  - Hardware bring-up
  - Silicon vendor tool chain support
  - FPGA and DSP programming
  - Simulation and emulation tools
  - Operating system and middleware configuration
  - Electronic Design Automation (EDA)

Questions?