eclipse Now You Can 2007

Eclipse Device Software Development Platform (DSDP)

Martin Oberhuber
Wind River
12 October 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)
- System Level Design
  - Virtual Prototyping Platform (VPP)

DSDP Overview

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

  *Create an **open, extensible**, scalable, and standards-based **development platform***

  *to address the needs of the **device software (embedded / mobile) 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 subprojects 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 0.9.6 released
- Dec 2006 TmL passes creation review, TM 1.0.1
- Jan 2007 eRCP 1.0.1, NAB 0.9.6-1
- June 2007 TM 2.0 and DD 0.9 released on Europa Train.
- Aug 2007 eRCP 1.1, Virtual Prototyping Platform (VPP) project proposed
- Sep 2007 TM 2.0.1

DSDP Leadership

Doug Gaff
PMC Lead

WIND RIVER

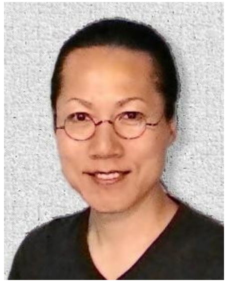
Pawel Piech
DD Lead

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

DSDP Stats

- 6 Projects – DD, eRCP, MTJ, NAB, TM, TmL and 1 proposed project VPP
- Over 550k Software Lines of Code (not counting comments)
- Over 40 committers representing (in alphabetical order):

Accelerated Technology A Mentor Graphics Division · ARM · ERICSSON · freescale semiconductor · FUJITSU · IBM · NOKIA Connecting People · palmsource · symbian · TEXAS INSTRUMENTS · WIND RIVER

- Other companies
  - Curtiss-Wright, Intel, QNX, 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)
- System Level Design
  - Virtual Prototyping Platform (VPP)

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), Ericsson, IBM, Mentor Graphics, Nokia, PalmSource, Symbian, TI, QNX, Freescale
- Completed during Callisto (June 06)
  - Build requirements and use cases for device software development needs in Eclipse.
  - Modify the Eclipse Debug Model Interfaces for customized embedded debugger implementations. (Released in Eclipse 3.2 as provisional API's.)
  - Enhance the platform memory view with embedded-specific renderings.

Device Debugging (DD)

- Europa Release – June 07
  - 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).
  - Preview IP-XACT editor and debugger views from SPIRIT consortium.

- Ganymede (currently planning) – June 08
  - Build a reference GDB/mi implementation for use with a GDB debug engine.
  - Enhance the debugger views for multi-core and multi-process support.
  - Productize IP-XACT Editor.
  - Provide a target description API based on IP-XACT.

DD – more detail

- The Eclipse 3.2 Debug Model (provisional API's)
  - 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
  - 1.0 release – June 08 on Ganymede train

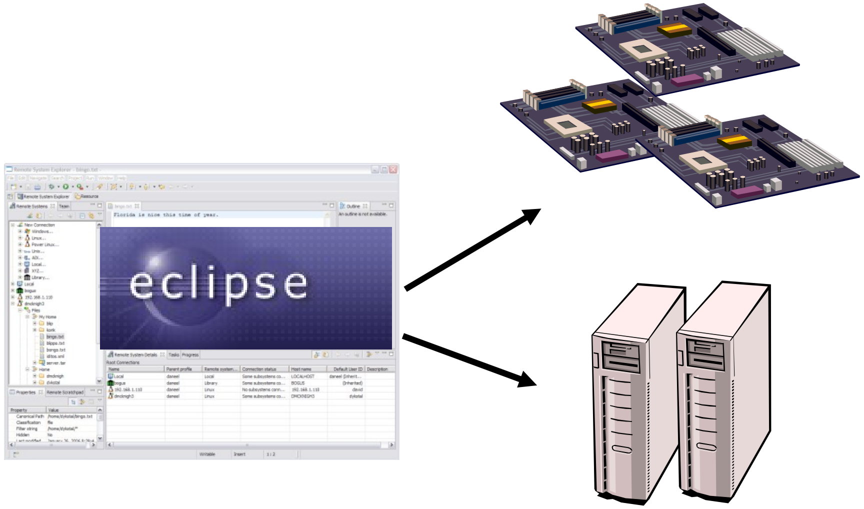
Target Management (TM)

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, QNX and many others

“Everything before Debugging” – Embedded to Enterprise Remote Access:

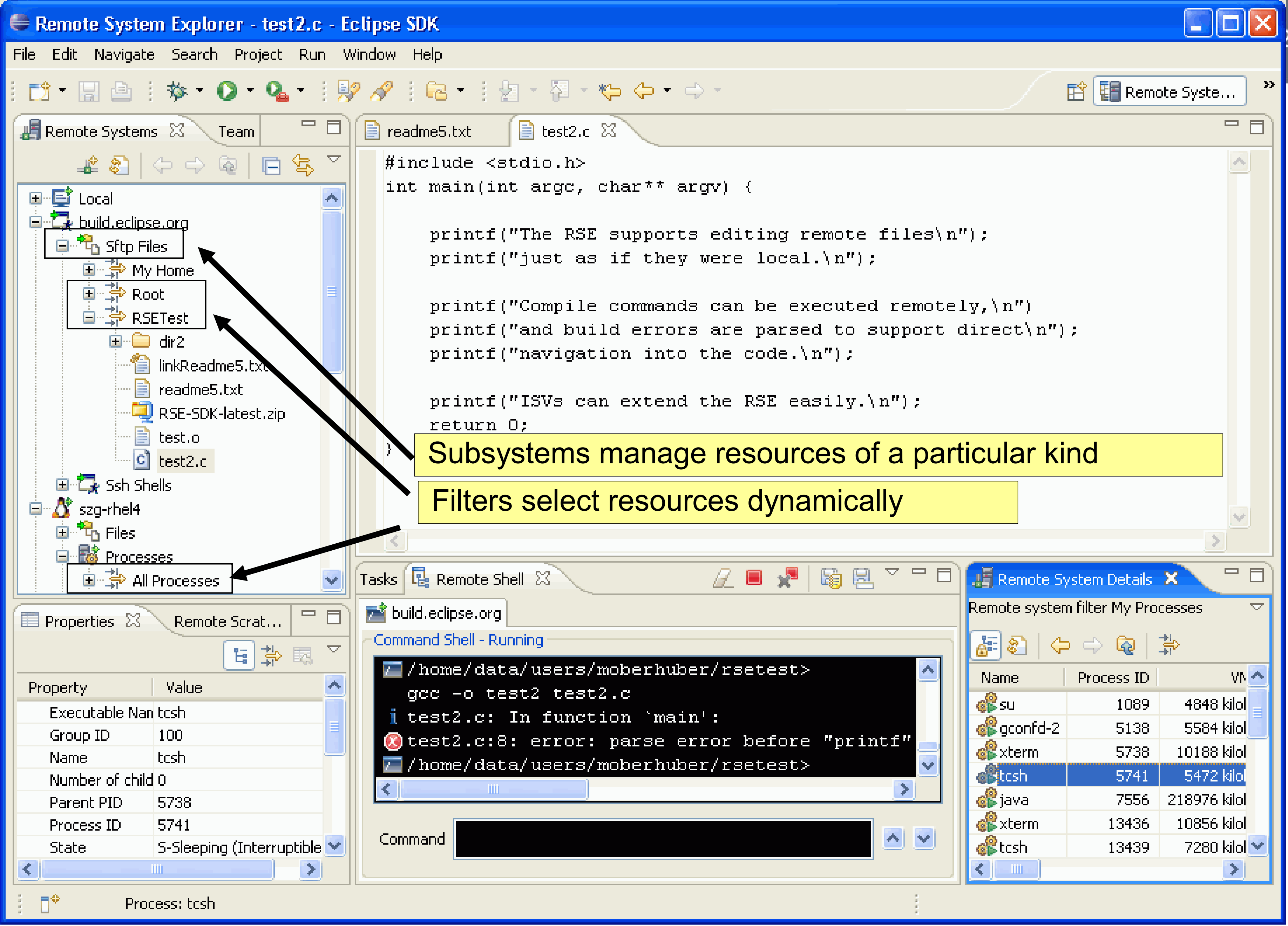
- *Team-shareable (standardized) connection data*
  - *(Future) Board Lab Management*
- *Pluggable, abstract connectivity*
  - *Discover and Drill Down*
- *A Meta-tool for plugged actions*
  - *Build, connect, get status*
  - *Download, run, debug, test*
  - *(Future) Scriptable Launch*



TM Features

- Features in 1.0.1 (December 06)
  - Remote System Explorer (RSE) Framework
    - transparent remote files, processes, shells
    - Dstore, SSH, FTP connection types
    - Integrate Jakarta Commons Net library for FTP access
  - CDT remote launch capabilities
  - Ansi terminal view & serial connection
  - Complete user and ISV documentation, tutorials and examples
  - Test on Windows, Linux, Solaris, Mac

- Small, independent components (Terminal, Discovery)
- And the RSE framework bringing it all together.



```
#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;
}
```

build.eclipse.org

Command Shell - Running

```
/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

Remote system filter My Processes

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

TM Features and Planning

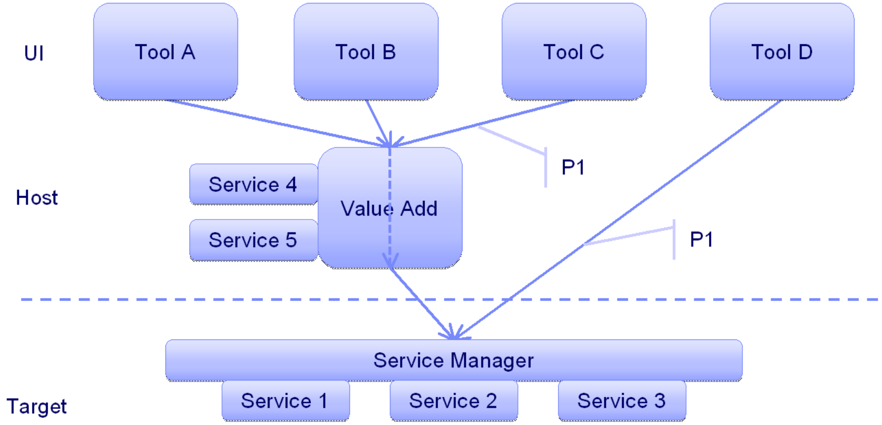
- Europa Release – June 07
  - Fix and improve the RSE EFS integration
  - Allow encoding of remote files to be specified
  - Contribute import/export from RSE7
  - Improve RSE SystemType and New Connection Wizard flexibility
  - Zeroconf / DNS-SD Discovery and Autodetect in RSE

- Ganymede (currently planning) – June 08
  - **Quality** – Reduce bug backlog, improve performance, API cleanup & hardening
  - **Scaling Down** - Improve UI/Non-UI splitting in RSE. Support RCP and headless launches
  - **Improve Persistence** Providers for import, export, migration of connection data
  - Improved **Remote CDT** Launch Integration
  - **Target Connection Framework (TCF)**
  - (proposed) Make RSE more dynamic and service-oriented
  - (proposed) Integrate the fast TM Terminal View with RSE
  - (proposed) Adopt Eclipse Platform 3.4 concepts, align with UI Guidelines
  - Further collaboration with other Eclipse projects (SoC WebDAV; Platform/Team Synchronization; TPTP; ECF; SWT deferred drag&drop)

Target Communication Framework (TCF)

- Today almost every device software development tool on the market has its own method of communication with target system.
  - Individual setup for each communication method
  - Especially awkward for multi-core (different tool for each core)

- The goal is a single protocol used to communicate between all tools and targets, supporting auto-discovery, multiplexing and tunneling
  - Transport protocol agnostic
  - Single point of configuration, single link



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)
- System Level Design
  - Virtual Prototyping Platform (VPP)

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, EclipseME project

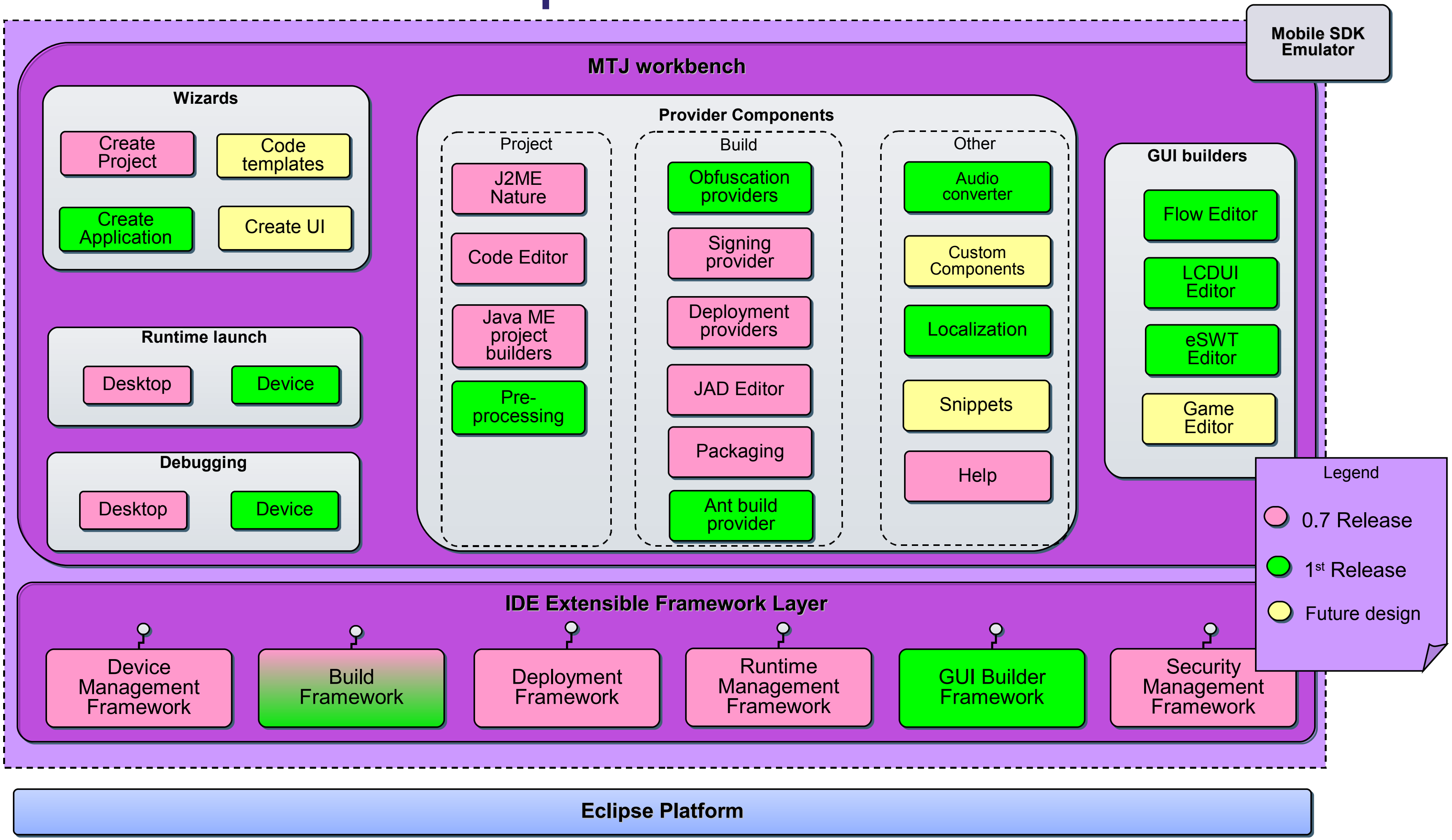
- Other participants
  - SonyEricsson, Sybase, Apogee Software, Sprint, Sysline Inc, Antenna, ShareME Technologies

- Release plans
  - 0.7 release in October 2006
  - 1.0 project plan in 3Q (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 Development Environment



Mobile SDK Emulator

MTJ workbench

Wizards
- Create Project
- Code templates
- Create Application
- Create UI

Runtime launch
- Desktop
- Device

Debugging
- Desktop
- Device

Provider Components

Project
- J2ME Nature
- Code Editor
- Java ME project builders
- Pre-processing

Build
- Obfuscation providers
- Signing provider
- Deployment providers
- JAD Editor
- Packaging
- Ant build provider

Other
- Audio converter
- Custom Components
- Localization
- Snippets
- Help

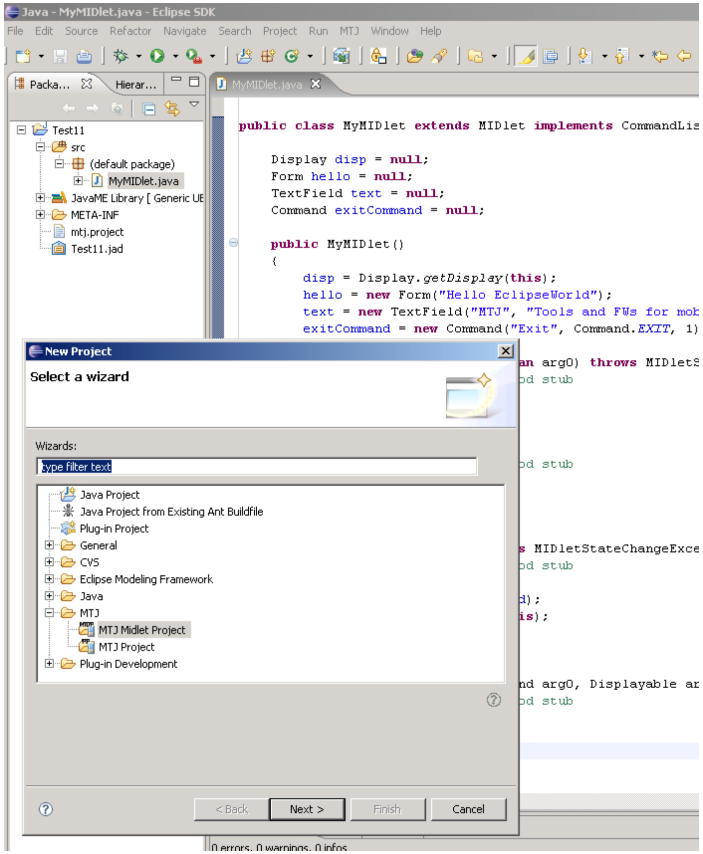
GUI builders
- Flow Editor
- LCDUI Editor
- eSWT Editor
- Game Editor

IDE Extensible Framework Layer
- Device Management Framework
- Build Framework
- Deployment Framework
- Runtime Management Framework
- GUI Builder Framework
- Security Management Framework

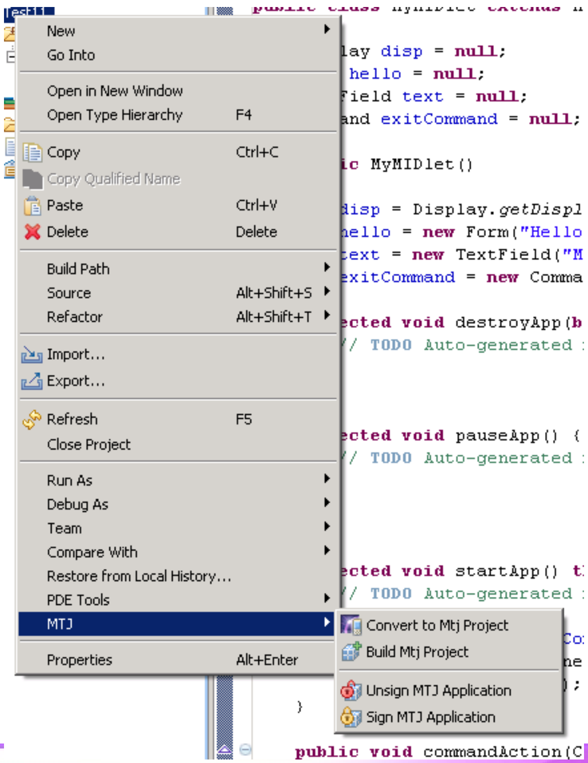
Eclipse Platform

Legend
- 🔴 0.7 Release
- 🟢 1st Release
- 🟡 Future design

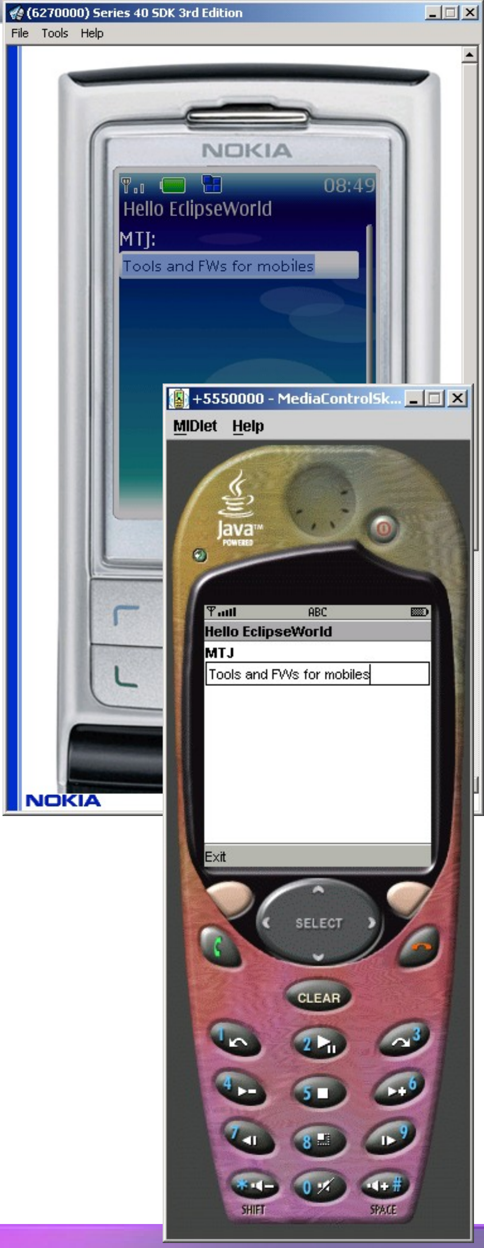
MTJ - Screenshots



```
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)

an arg0) throws MIDletS
od stub

od stub

s MIDletStateChangeExce
od stub
d);
is);

nd arg0, Displayable ar
od stub
```



```
public class MyMIDlet extends M
lay disp = null;
hello = null;
Field text = null;
and exitCommand = null;

ic MyMIDlet()

disp = Display.getDispl
hello = new Form("Hello
ext = 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 :
Co
ne
);
}
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
- 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 1.1 (Aug 2007)
  - Nokia Series 80
  - Windows Desktop
  - Windows Mobile
  - WinCE 5.0 Professional
- Future Release
  - Nokia S60
  - Linux Qte, Linux GTK
  - UIQ

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)
- System Level Design
  - Virtual Prototyping Platform (VPP)

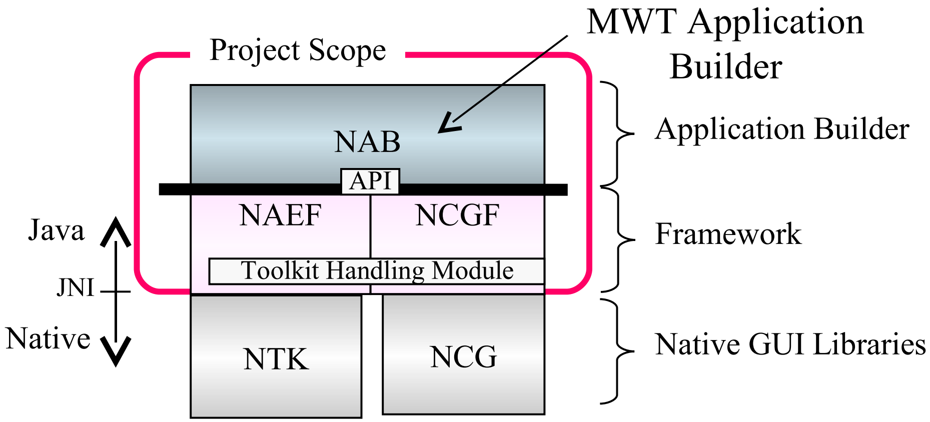
Native Application Builder (NAB)

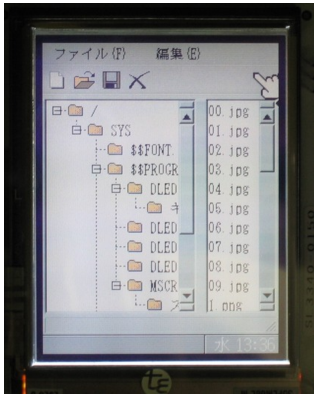
www.eclipse.org/dsdp/nab

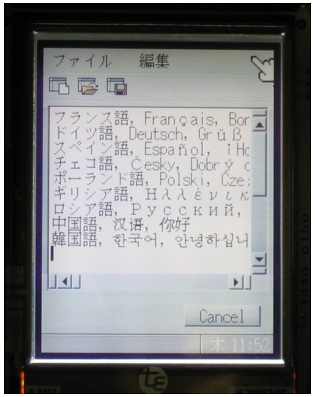
- Mission: *Create a C++ GUI builder for embedded operating systems, similar to eSWT for Java.*
- Fujitsu (lead), WideStudio team, Eclipse Japan Working Group
- Features
  - Visual editor for building GUI's
  - C++ application framework
  - MWT (Multiplatform Widget Toolkit) – separate download

Supported Platforms of MWT	
OS	Windows, Linux, MacOSX, FreeBSD, Solaris, T-Engine, ITRON, BTRON
CPU	IA32, ARM, SH3/4, FRV, MIPS, PPC, SPARC
Graphics	X11 Server, DirectFB, Frame Buffer, Win32, WinCE, T-Engine(T-Shell), MacOS

NAB: Architecture and Sample Applications







フランス語, Français, Bor
ドイツ語, Deutsch, Grüß
スペイン語, Español, ¡Ho
チェコ語, Česky, Dobrý c
ポーランド語, Polski, Cze:
ギリシア語, Ηλλένικ
ロシア語, Русский,
中国語, 汉语, 你好
韓国語, 한국어, 안녕하십니

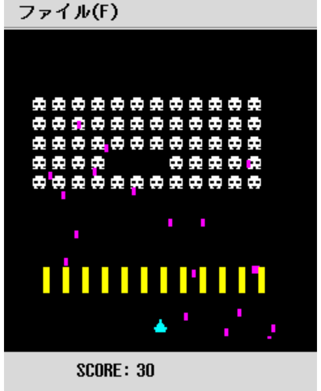
NAB: Native Application Builder

NAEF: Native Application Editor Framework

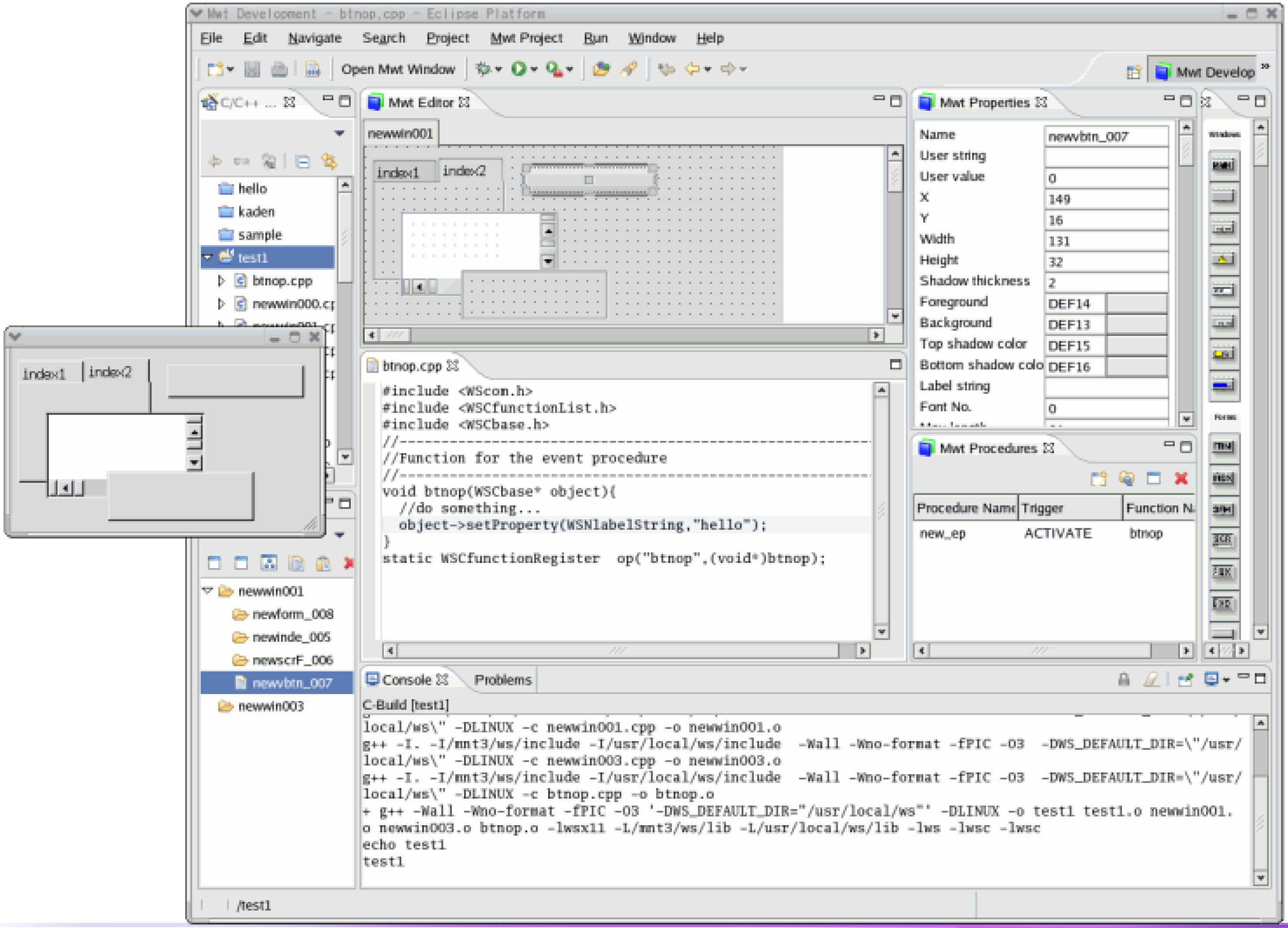
NCGF: Native Code Generation Framework

NCG: Native Code Generator

NTK: Native Tool Kit



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);
```

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

Procedure Nam	Trigger	Function N
new_ep	ACTIVATE	btnop

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/dsdp/tml

- Motorola (lead)
- Create 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
    - Mobile device simulation
  - 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
  - Gathering community and building initial development team
  - Mobile Linux Emulator Framework
    - Generic framework to support different device emulator architectures
    - VMware, User-mode Linux (UML) emulators, QEmu emulators, etc.
    - VNC Viewer integrated with TM project
  - Simulated end-to-end environment
    - Emulated devices, simulated services, simulated network nodes

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)
- System Level Design
  - Virtual Prototyping Platform (VPP)

Proposed Project

Virtual Prototyping Platform (VPP)

www.eclipse.org/proposals/vpp

- Proposed Project
  - *Enable Eclipse to be used for the tasks associated with constructing, debugging, visualizing, analyzing, and using (models of) systems constructed from components that may be based on hardware or software, or a combination of both.*
- Project Scope
  - Model Debug
  - System Visualization
  - Model specific Analysis
  - Transaction level analysis (communication between Virtual components)
  - Model execution profiling
  - User interactions
  - Configuration
  - Control (User based control, and script based control)
- Initial technologies
  - SystemC Syntax highlighting
  - SystemC topology viewer (to view the hardware components of a system, typically fed by a SystemC parser)
  - Display of analysis information published by SystemC models
  - Profiling of SystemC simulations (this requires hooks into the SystemC kernel to provide simulation activity in order to improve simulation performance)

VPP continued

- Interested Parties
  - Industrial Organizations
    - GreenSocs
    - Xilinx Research
    - (others to be announced soon)
  - University Organizations
    - FZI (a research organization of German Universities)
    - Unicamp (a similar organization based in Brazil)
    - EIS (a research lab at TU Braunschweig, Germany)
- Roadmap
  - The intention is to provide basic plugins to support SystemC modeling and debug straight away.
  - These will not be of sufficient quality to warrant a 1.0 release, but will be worked on to achieve that goal.
  - Other parallel activities will focus on model to user interactions.
  - This work is expected to be industrially funded, with GreenSocs providing the resource.

Questions?

For more information, go to http://www.eclipse.org/dsdp
Links to all project downloads, FAQs, newsgroups, tutorials, …