

Device Software Development Platform (DSDP) Project

Doug Gaff
Wind River Systems
12 October 2006

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 functions as a container for “all things embedded” in Eclipse.

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 and MTJ 0.7 to be released
- [Jan 2006 – today](#) Milestone releases of technology in sub-projects.

DSDP Stats

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



- 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)

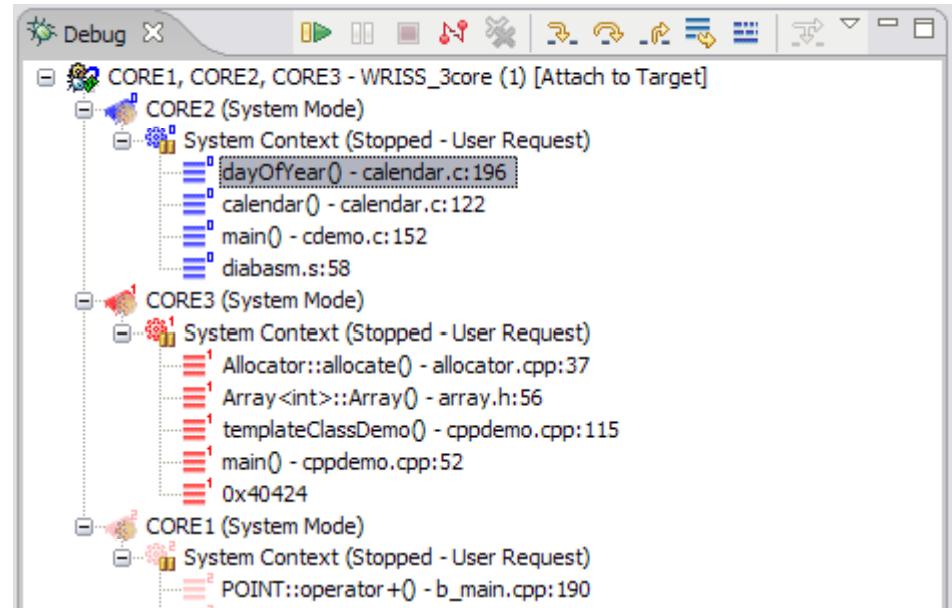
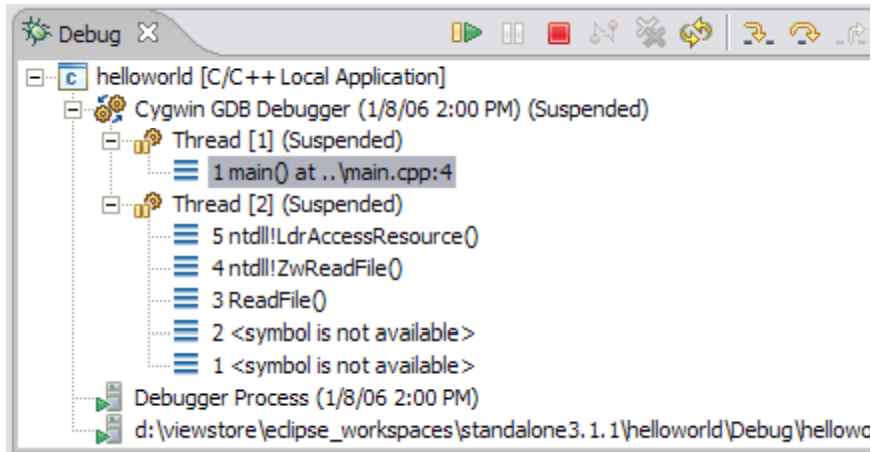
www.eclipse.org/dsdp/dd

- 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) . – in progress.
 - 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.

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
 - 0.9 release – June 07 with Europa train

DD: Eclipse 3.1 vs. Eclipse 3.2+ Debug Model



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

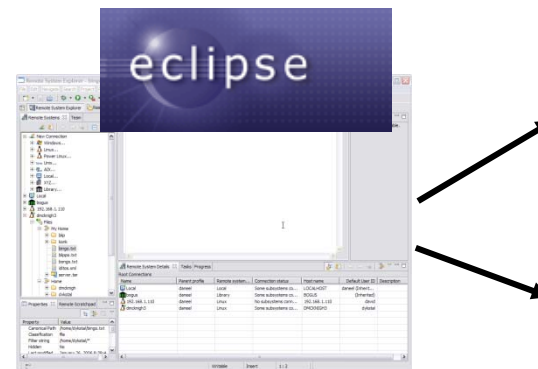
- Mission: **Create data models and frameworks to configure and manage embedded systems, their connections, and their services.**
- Wind River (lead), IBM, LANL, MontaVista, PalmSource, Symbian
- Freescale, Mentor Graphics, Nokia, TI, QNX

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*



TM Features

- Features for 1.0 (October 2006)
 - 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
 - Test on Windows, Linux, Solaris, Mac
 - Zeroconf Discovery, EFS Integration – Preview Release

TM Future Plans (proposed)

- 2.0 release in June 07 (Europa train)
 - More Standards based target connections (Telnet, ECF)
 - Terminal Emulation Framework and View
 - User-Defined Actions, Import/Export
 - Component-Based Launching (CBL)
 - Multi-core / Multi-target support through connection groups
 - Connection Model for HW Debugging (complex connector setup)
- Beyond 2.0
 - Flexible Target Connector framework, Connector plumbing algorithm
 - Adapters for Target access control (shared board labs)
 - SSH tunneling connector

The screenshot displays the Eclipse IDE interface for Remote System Explorer. The main window is titled "Remote System Explorer - test2.c - Eclipse SDK". The interface is divided into several panes:

- Remote Systems:** A tree view on the left showing a hierarchy of remote systems. The selected system is "build.eclipse.org", which contains a "Sftp Files" subsystem. Under "Sftp Files", there is a "My Home" subsystem, which in turn contains a "Root" subsystem and an "RSETest" subsystem. The "RSETest" subsystem contains a "dir2" folder and several files, including "test2.c".
- Code Editor:** The central pane shows the source code of "test2.c". The code includes a standard C main function with several printf statements. A yellow callout box points to the code with the text: "Subsystems manage resources of a particular kind".
- Remote Shell:** A terminal window at the bottom center shows the execution of the program. The prompt is "/home/data/users/moberhuber/rsetest>". The user enters "gcc -o test2 test2.c", followed by "i test2.c: In function 'main':", and then a red error message: "x test2.c:8: error: parse error before 'printf'". A yellow callout box points to the error message with the text: "Filters select resources dynamically".
- Remote System Details:** A pane on the bottom right shows a table of processes running on the remote system. The table is titled "Remote system filter My Processes" and has columns for Name, Process ID, and VM. The processes listed are: 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), and tcsh (13439, 7280 kilol). The "tcsh" process with PID 5741 is highlighted.
- Properties:** A pane on the bottom left shows the properties of the selected process (tcsh). The properties include: Executable Name (tcsh), Group ID (100), Name (tcsh), Number of child (0), Parent PID (5738), Process ID (5741), and State (S-Sleeping (Interruptible)).

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 in October
 - 1.0 project plan for Europa in progress

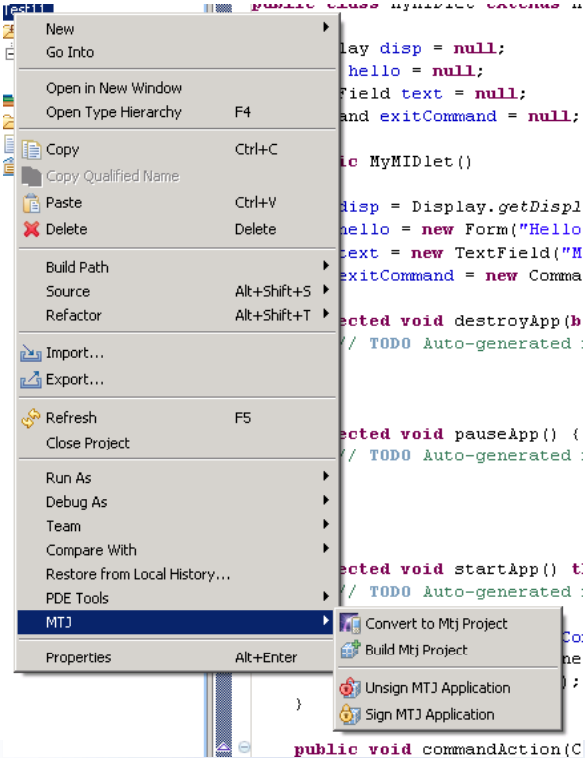
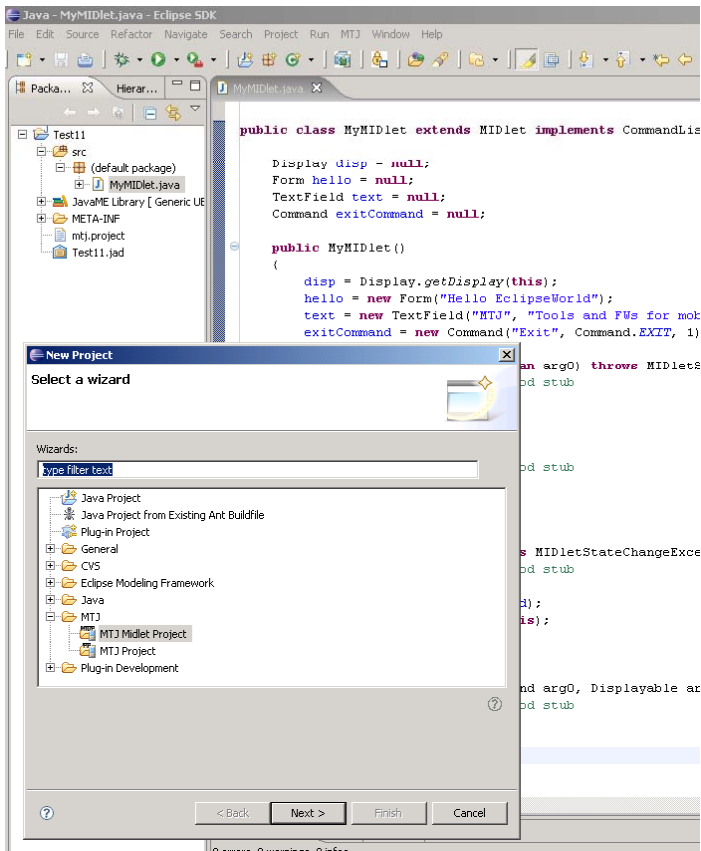
MTJ 0.7 Features (Oct 2006)

- To create Eclipse Mobile Java Tools platform that vendors can extend to support their devices. Extensibility in the first phase 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



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 (Sept 22)
 - 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

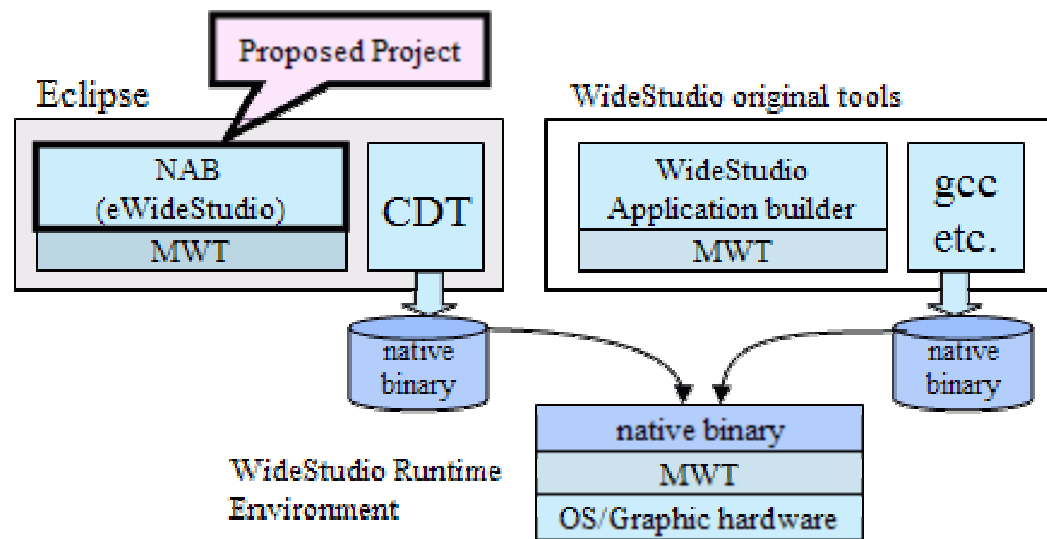
- DSDP Overview
- General embedded support
 - Device Debugging (DD)
 - Target Management (TM)
- Mobile Java
 - Mobile Tools for the Java Platform (MTJ)
 - Embedded Rich Client Platform (eRCP)
- **Mobile C/C++**
 - Native Application Builder (NAB)
 - Tools for Mobile Linux (TmL)
- Getting Involved

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
 - Pre-release currently available
 - 1.0 project plan in progress (Europa train)

NAB: Architecture

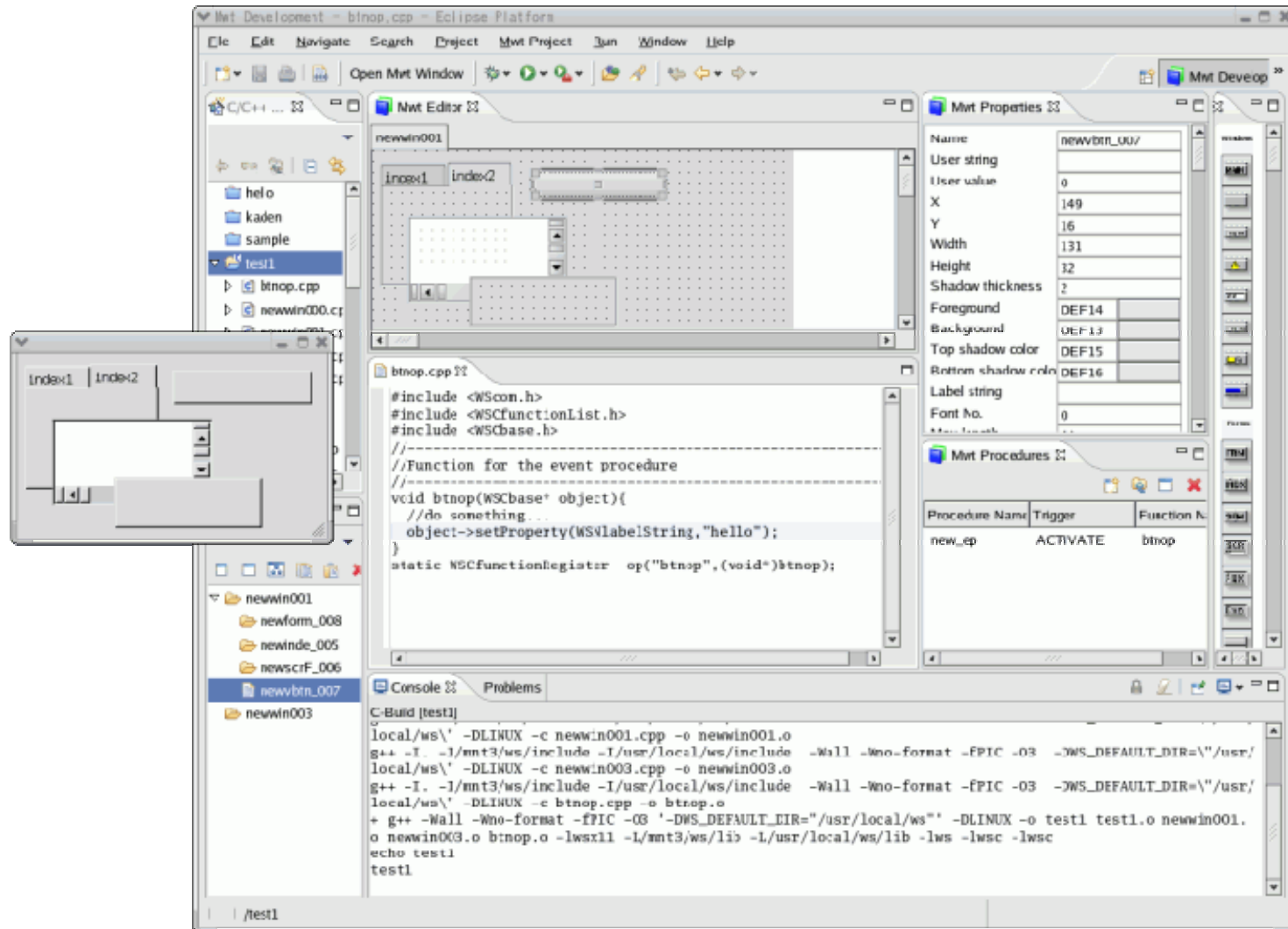


NAB Provides

- 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



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
- Release plans
 - In proposal phase
 - 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
 - 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)

