



TmL Use Cases

DSDP/TmL: Tools for mobile Linux (TmL)

Revision History

Revision	Description	Modified by	Date
01.00-D01	Initial version	Fábio Fantato	09 Jan, 2007
01.00-D02	Refinements according to latest discussions	Mauren Brenner	18 Jan 2007

VERSION: 00.01-D02

Distribution List

Name	Company	Location	Role
Fábio Fantato	Motorola	wfr004@motorola.com	Author
Mauren Brenner	Motorola	wmb058@motorola.com	Approver
Christian Kurzke	Motorola	e11581@motorola.com	Approver
DSDP-PMC	Eclipse		Approver

Index

1. ABOUT THIS DOCUMENT	3
2. SCOPE	3
2.1. PROJECT GOALS	3
2.2. INITIAL SCOPE FOR TML	4
2.3. SUPPORT TO MULTIPLE BUILD TARGETS PER PROJECT	4
2.4. SUPPORT TO MOBILE DEVICE EMULATORS.....	4
2.5. SIMULATED END-TO-END ENVIRONMENT.....	6
3. USE CASES	6
3.1. SUPPORT TO MULTIPLE BUILD TARGETS.....	6
3.1.1 ACTORS	6
3.1.2 VENDOR USE CASES	6
3.1.2.1 CREATE A BUILD TARGET	6
3.1.2.2 MODIFY A BUILD TARGET.....	7
3.1.3 DEVELOPER USE CASES	7
3.1.3.1 INSTALL A BUILD TARGET	7
3.1.3.2 CREATE A NEW PROJECT WITH MULTIPLE BUILD TARGETS	7
3.1.3.3 ADD A NEW BUILD TARGET TO AN EXISTING PROJECT.....	8
3.1.3.4 REMOVE A BUILD TARGET FROM AN EXISTING PROJECT	8
3.1.3.5 UPDATE A BUILD TARGET IN AN EXISTING PROJECT.....	8
3.1.3.6 UPDATE A BUILD TARGET FOR THE ENTIRE WORKSPACE	9
3.1.3.7 BUILD A PROJECT FOR A SPECIFIC BUILD TARGET	9
3.1.3.8 BUILD A PROJECT FOR ALL BUILD TARGETS.....	9
3.1.3.9 SPECIFY THE BUILD TARGETS FOR AUTOMATIC BUILD	10
3.1.3.10 AUTOMATICALLY BUILD PROJECT FOR SPECIFIED TARGETS.....	10
3.2. SUPPORT TO MOBILE DEVICE EMULATORS.....	10
3.2.1 ACTORS	10
3.2.2 VENDOR USE CASES	11
3.2.3 DEVELOPER USE CASES	11
3.2.3.1 INSTALL A DEVICE EMULATOR.....	11
3.2.3.2 CONFIGURE A DEVICE EMULATOR.....	11
3.2.3.3 START A DEVICE EMULATOR.....	11
3.2.3.4 DEPLOY AN APPLICATION ONTO A DEVICE EMULATOR	12
3.2.3.5 START A DEPLOYED APPLICATION ON A DEVICE EMULATOR	12
3.2.3.6 STOP AN APPLICATION ON A DEVICE EMULATOR	12
3.2.3.7 PROVIDE INPUT TO A DEVICE EMULATOR	12
3.2.3.8 VISUALIZE OUTPUT FROM A DEVICE EMULATOR	13
3.3. SIMULATED END-TO-END ENVIRONMENT.....	13
3.3.1 ACTORS	13
3.3.2 DEVELOPER USE CASES	13
3.3.2.1 INSTALL A SIMULATED SERVER.....	13
3.3.2.2 ENABLE A DEVICE TO WORK ON THE SIMULATED END-TO-END ENVIRONMENT	13
3.3.2.3 ENABLE A DEVICE EMULATOR TO WORK ON SIMULATED ENVIRONMENT	14
3.3.2.4 CONNECT ELEMENTS OF THE SIMULATED END-TO-END ENVIRONMENT	14
4. REFERENCES	14

1. About this document

This document contains the use cases for the initial scope proposed for the Tools for mobile Linux (TmL) project.

2. Scope

This section summarizes the proposal and initial scope for the TmL project.

2.1. Project Goals

The main goal of the TmL project is to provide frameworks and exemplary tools that collectively support the entire software development cycle as described by the keywords *Define*, *Design*, *Develop*, *Debug*, and *Deploy* and depicted in **Figure 1** below.

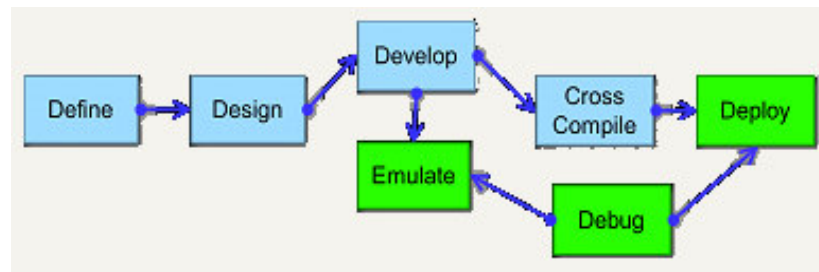


Figure 1 Software development cycle

- *Define*:
 - system software architecture
- *Design*:
 - software modeling
 - GUI design
- *Develop*:
 - coding
 - cross compilation and toolchains
 - managed make to automate build process
- *Debug*:
 - device emulation
 - on-device debugging
 - unit and regression testing on device and emulator
 - profiling
 - testing of enterprise applications on simulated end-to-end environment
- *Deploy*:
 - application packaging and signing
 - deployment into device or emulator

2.2. Initial Scope for TmL

The initial scope for TmL is described in the project proposal [1], and is divided into three major areas:

- Support to multiple build targets per project;
- Support to mobile device emulators;
- Simulated end-to-end environment for testing of enterprise applications.

Requirements for each one of the major areas are outlined below.

2.3. Support to Multiple Build Targets per Project

Support to cross-compilation for multiple targets is a fundamental feature of any development environment of native C/C++ applications for embedded devices. An application developer will often want to *compile the same application for several targets*, e.g. when a new release of a device becomes available.

The multiple build target mechanism must satisfy the following requirements:

- It must be possible to specify all the tools and utilities required to build an application for a given target in the corresponding build target.
- It must be possible to specify all the libraries and fixed inputs required to build an application for a given target in the corresponding build target.
- It must be possible to specify build rules for a build target that completely specify how the source code inputs and the fixed inputs provided in the build target package shall be processed by the tools and utilities in order to produce the final outputs of the build process. The final outputs of the build process must contain all the components that are necessary to deploy the corresponding application to its target device.
- The developer must be able to specify one or more build targets when a new project is created.
- The developer must be able to specify new build targets and/or remove build targets after the project is created.
- It must be possible to build the (same) application project for each one of the applicable build targets, thus creating the final deployable artifacts.
- It must be possible to install and/or remove build targets from the system independently of one another and independently of those provided by CDT by default; in other words, build targets must not interfere with one another.

2.4. Support to Mobile Device Emulators

An important tool for development of applications for mobile devices is a device emulator. This is indeed a fundamental piece in cases where a device prototype is not yet available for internal software developers, or where the device has not yet been released to

independent or 3rd party developers, but it can also be useful in several other situations, e.g. when a developer is evaluating the portability of an application to a different device, and so on. In this context, an emulator can be defined as a software component that runs on the host system and that is capable of executing software components developed for the target platform.

Emulators are especially useful for functional testing and debugging of applications, while other tasks, notably profiling and performance optimization, typically require the availability of the actual target device.

A generic emulator framework involves components on the development environment side as well as on the actual emulator side. The components on the device and emulator side will be in most cases specific implementations, while the components on the IDE side can and should be as common and reusable as possible. The TmL project intends to address this issue by providing specifications of *interfaces* and *protocols* whereby the software components (or "agents") on the device or emulator and those on the IDE can communicate in order to achieve their purposes.

The emulator framework must satisfy the following requirements:

- It must be general enough to support different kinds of emulators, including:
- A processor emulator that emulates the target processor and is capable of executing its binary instructions (machine code), being thus capable (in principle) of executing any software component developed for the target device;
- Higher-level emulators that essentially provide a virtual machine that corresponds to a certain level of software and hardware of the target device; for instance, one such emulator could provide a virtual machine that comprises emulated hardware, operating system and libraries, and is thus capable of executing user-level applications developed for the target device, but cannot execute any system-level components beside those that are already part of the emulator;
- An emulator at an even higher level; for instance, an emulator that provides a complete Java virtual machine and is therefore capable of executing Java applications but not native or system software.
- It must be possible to deploy applications to the emulator;
- It must be possible to run the applications deployed to the emulator;
- It must be possible to debug applications on the emulator.
- It must include well-defined protocols that can be used as guidelines for emulator developers that wish to integrate their emulators into the Eclipse platform using the TmL emulator framework;
- It must provide Eclipse views that allow the developer to view device output, monitor emulator status and provide input to the emulator.

2.5. Simulated End-to-End Environment

Enterprise applications unleash the actual potential of mobile devices by exploiting their network connectivity. Testing of such applications may require complex and expensive setting up of networks involving specialized servers, such as SMS, MMS and so on. A definite asset of a development environment is a simulated end-to-end environment that includes simulated servers of standard services, allowing the developer to connect devices or emulators to run and test enterprise applications.

The framework for simulation of end-to-end environment must satisfy the following requirements:

- It must be possible to connect devices and device emulators to simulated servers.

3. Use Cases

3.1. Support to Multiple Build Targets

3.1.1 Actors

Vendor

A vendor is a mobile device manufacturer. Vendors typically create and distribute the toolchains, libraries, header files and other artifacts required to compile applications for their platforms.

Developer

A developer is a user of the development environment and/or tools. Developers typically use the toolchains, libraries and other artifacts provided by vendors to create applications for their respective platforms.

3.1.2 Vendor Use Cases

3.1.2.1 Create a Build Target

- Actor:** Vendor.
- Summary:** Support to creation of a build target.
- Description:** Allow vendor to specify name, description, rules, tools, dependencies, activation order and valid arguments for the tools and all required artifacts for a build target.
- Result:** A new plug-in that contains a build target definition that can be installed on the developer's Eclipse environment.

3.1.2.2 Modify a Build Target

Actor: Vendor.
Summary: Support to modification of a build target.
Description: Allow vendor to modify any property of a build target.
Result: A new version of the build target plug-in.

3.1.3 Developer Use Cases

3.1.3.1 Install a Build Target

Actor: Developer.
Summary: Support to installation or update of a build target plug-in.
Description: Allow developer to install or update a build target plug-in containing all that is needed to compile applications for the corresponding device or emulator.
Result: The new build target must be installed or updated without affecting any of the other build targets on the system; the new build target must be available for selection when creating new projects or when modifying the properties of existing projects.

3.1.3.2 Create a New Project with Multiple Build Targets

Actor: Developer.
Summary: Support to creation of new projects selecting one or more build targets.
Description: Allow developer to create new projects by including in the New Project Wizard the option whereby they can select the build targets that apply to the project. Initial content for each project include configuration files, makefiles and other build rules customized for the project.
Result: The new project is open and supports the selected build targets.

3.1.3.3 Add a New Build Target to an Existing Project

- Actor:** Developer.
- Summary:** Support the addition of a new build target to an existing project.
- Description:** Allow developer to add a build target to an existing project, creating all the configuration files, makefiles and other build rules needed by the build target.
- Result:** The project now supports the new build target in addition to those that it already supported.

3.1.3.4 Remove a Build Target from an Existing Project

- Actor:** Developer.
- Summary:** Support the removal of a build target from an existing project.
- Description:** Allow developer to remove a build target from an existing project, removing all the files related to that target from the project directory.
- Result:** The project no longer supports the removed build target and all files related to the target are no longer in the project directory, while other targets are not affected.

3.1.3.5 Update a Build Target in an Existing Project

- Actor:** Developer.
- Summary:** Support the update of a build target in an existing project.
- Description:** Allow developer to update a build target in an existing project, replacing the old configuration, makefiles and other build rules with those from the new version of the build target plug-in.
- Result:** The build target is still associated to the project but all the build target structure for that target in the project directory is updated, without affecting any of the other targets.
- Comments:** Updating a build target means substituting a new build target for some old build target in a project. The old and new build targets are *not* different versions of the same plug-in; they are *different* plug-ins and represent e.g. different *device* versions.

3.1.3.6 Update a Build Target for the Entire Workspace

- Actor:** Developer.
- Summary:** Support the update of a build target for all projects in the current workspace.
- Description:** Allow developer to update a build target in all projects to which the build target is associated.
- Result:** The build target is still associated to all the projects to which it was associated but all the build target structure for that target in all project directories in the workspace is updated, without affecting any of the other targets.

3.1.3.7 Build a Project for a Specific Build Target

- Actor:** Developer.
- Summary:** Support compilation of a project's source files and generation of executable file(s) for a specific build target.
- Description:** Allow developer to select a specific build target and generate (or regenerate) the executable file(s) for a project for the selected build target.
- Result:** Executable file(s) is (are) generated or regenerated for the project for the selected build target, without affecting any of the other targets.

3.1.3.8 Build a Project for All Build Targets

- Actor:** Developer.
- Summary:** Support compilation of a project's source files and generation of executable files for all build targets associated to the project.
- Description:** Allow developer to generate (or regenerate) the executable files for a project for all associated build targets.
- Result:** Executable files are generated or regenerated for the project for all the associated build targets.

3.1.3.9 Specify the Build Targets for Automatic Build

- Actor:** Developer.
- Summary:** Support selection of zero or more build targets associated to a project so that executable files for the selected targets are generated (or regenerated) when the Automatic Build option is enabled and the project is automatically built.
- Description:** Allow developer to select zero or more build targets among those associated to a project so that the executable files for the selected targets are generated when the project is automatically built.
- Result:** Whenever the project is automatically built using the Automatic Build feature of Eclipse, executable files are generated (or regenerated) for the selected build targets, without affecting any of the other targets.

3.1.3.10 Automatically Build Project for Specified Targets

- Actor:** Developer.
- Summary:** Support enabling of Automatic Build option for projects that support multiple build targets.
- Description:** Allow developer to specify that a project must be automatically built using the Eclipse Automatic Build mechanism.
- Result:** The Automatic Build mechanism will be triggered for the project for all the build targets specified by the developer.

3.2. Support to Mobile Device Emulators

3.2.1 Actors

Vendor

Vendors are mobile device manufacturers. Vendors typically create and distribute emulators for the devices they produce.

Developer

Developers are the users of the development environment and/or tools. Developers typically use device emulators to test and debug the applications they develop.

3.2.2 Vendor Use Cases

Support to creation and packaging of device emulators is not yet planned. For one thing, device vendors may have completely different emulator structures, so it is not yet clear whether it is advantageous or not to have tools to support emulator creation by vendors.

3.2.3 Developer Use Cases

3.2.3.1 Install a Device Emulator

Actor: Developer.
Summary: Support to installation or update of a device emulator.
Description: Allow developer to install or update a device emulator plug-in containing all that is necessary to run the emulator and deploy, run and debug applications on it.
Result: The device emulator must be installed or updated without affecting any of the other device emulators on the system; the new emulator must be available for selection when deploying applications compiled for compatible build targets.

3.2.3.2 Configure a Device Emulator

Actor: Developer.
Summary: Support to definition or update of the configuration of a device emulator, e.g. network (IP) address.
Description: Allow developer to configure any applicable parameters and properties of a device emulator.
Result: The new configuration must be applied to all new instances of the device emulator that are started after the changes.

3.2.3.3 Start a Device Emulator

Actor: Developer.
Summary: Support the starting of a device emulator.
Description: Allow developer to start a device emulator.
Result: The device emulator must be started and ready for applications to be deployed onto it.

3.2.3.4 Deploy an Application onto a Device Emulator

- Actor:** Developer.
- Summary:** Support the deployment of an application onto a device emulator.
- Description:** Allow developer to take executable file(s) generated for a specific build target and deploy it (them) onto a compatible device emulator, if available.
- Result:** The executable file(s) must be transferred to the device emulator and unpackaged and/or installed so that the application can be executed.

3.2.3.5 Start a Deployed Application on a Device Emulator

- Actor:** Developer.
- Summary:** Support the execution of a deployed application on a device emulator.
- Description:** Allow developer to start execution of an application that has been previously deployed onto the device emulator.
- Result:** The application must be started on the device emulator.

3.2.3.6 Stop an Application on a Device Emulator

- Actor:** Developer.
- Summary:** Support the stopping of an application on a device emulator.
- Description:** Allow developer to stop the execution of an application that is running on the device emulator.
- Result:** The application must be stopped on the device emulator.

3.2.3.7 Provide Input to a Device Emulator

- Actor:** Developer.
- Summary:** Support input to a device emulator, e.g. keystrokes.
- Description:** Allow developer to use the development environment to send specific input events to a device emulator.
- Result:** The input events must be sent to the device emulator as if they were produced on the actual device.

3.2.3.8 Visualize Output from a Device Emulator

- Actor:** Developer.
- Summary:** Support visualization of output from a device emulator, e.g. device display.
- Description:** Allow developer to use the development environment to visualize output from a device emulator.
- Result:** Output must be sent from the device emulator to the development environment and displayed in a way that is meaningful to the developer.

3.3. Simulated End-to-End Environment

3.3.1 Actors

Developer

Developers are the users of the development environment and/or tools. Developers of enterprise applications use the simulated end-to-end environment in conjunction with appropriate devices and/or device emulators to run and test the applications.

3.3.2 Developer Use Cases

3.3.2.1 Install a Simulated Server

- Actor:** Developer.
- Summary:** Support installation of a simulated server on the development environment.
- Description:** Allow developer to install a simulated server plug-in on the development environment.
- Result:** The simulated server must be available as a component of the simulated end-to-end environment.

3.3.2.2 Enable a Device to work on the Simulated End-to-End Environment

- Actor:** Developer.
- Summary:** Support enabling of a device to work on the simulated end-to-end environment.
- Description:** Allow developer to install software agents on the device to enable it to work on the simulated end-to-end environment, i.e. communicate with other elements where applicable.
- Result:** The device can be connected to other elements of the simulated end-to-end environment.

3.3.2.3 Enable a Device Emulator to work on Simulated Environment

Actor: Developer.

Summary: Support enabling of a device emulator to work on the simulated end-to-end environment.

Description: Allow developer to install software agents on the device emulator to enable it to work on the simulated end-to-end environment, i.e. communicate with other elements where applicable.

Result: The device emulator can be connected to other elements of the simulated end-to-end environment.

3.3.2.4 Connect Elements of the Simulated End-to-End Environment

Actor: Developer.

Summary: Support connection of elements of the simulated end-to-end environment, e.g. appropriate devices, device emulators and simulated servers.

Description: Allow developer to connect elements of the simulated end-to-end environment to one another.

Result: The connected elements will allow the developer to run and test enterprise applications.

4. References

[1] TmL Project Proposal: <http://www.eclipse.org/proposals/tml>