

LIGHTWEIGHT USER INTERFACE TOOLKIT: ADVANCED USER INTERFACES FOR MASS-MARKET PHONES

ADVANCED USER INTERFACES FOR MASS-MARKET DEVICES

KEY FEATURES

- Small footprint library - can be easily bundled with the application
- Modern user interface capabilities such as animations, transitions, layouts, UI controls, 3D capabilities, etc
- Swing-inspired programming model provides full control of component appearance, including theming and pluggable look-and-feel
- Forms, lists and controls can be displayed consistently across most Java ME enabled handsets
- Low dependency on Java stack functionality, requiring only CLDC 1.1 and MIDP 2.0
- Support for modern hardware including 3D acceleration (via JSR 184) and touchscreen input
- Mobile web content can be rendered seamlessly with HTMLComponent
- Support for prototyping and debugging LWUIT applications in the Java ME SDK

KEY BENEFITS

- Provides modern user interface capabilities that can be deployed to mass-market devices
- Bundles with your application, enabling you to deploy LWUIT applications to existing, in-market devices
- Theme Creator enables rapid customizing of LWUIT user interfaces

With the proliferation of mobile handsets based on Java™ Platform, Micro Edition (Java ME) technology, creating applications that can easily run across all of these devices can be a challenge. Too often, the developer's assumptions about user interface and layout are different from device to device. To help solve this problem, Oracle has developed the Lightweight User Interface Toolkit (LWUIT) for Java ME enabled handsets.



Advanced User Interfaces for Mass-market Devices

Smart User Interface Technology for Everyone

LWUIT offers compelling UI capabilities and a clean API that is inspired by Swing. To breathe life into their applications, developers can leverage built-in transitions and other visual effects. Developers can also build their own UI components or leverage included components to provide a consistent and compelling look and feel to their applications, which can then run across a wide range of devices across multiple vendors. In addition, LWUIT supports 3D rendering through JSR 184 and support for Scalable Vector Graphics

Portability and Broad Device Support

Applications built with LWUIT can present a rich and consistent user interface across multiple devices and can automatically adapt to and takes advantage of device-specific properties such as screen size, graphics capabilities, and touch screen support. To achieve these portability improvements, LWUIT was built using low-level common elements in MIDP 2.0 (or similar basic graphics capabilities on other platforms). The LWUIT layout manager also helps achieve portability by helping applications easily adapt to screen-size differences

In addition, LWUIT is continually tested across a wide range of today's mass market devices - from low-end phones with limited memory, small screens, and numeric keypads all the way to high-end devices with fast processors, high-resolution touch screens and built-in keyboards.

Programming with the LWUIT APIs

LWUIT allows the display of any collection of user interface controls, in a presentation

analogous to the LCDUI forms mechanism in MIDP. However, unlike LCDUI, all rendering is done on a MIDP canvas to ensure wide portability. LWUIT offers methods for showing information on forms and receiving user input. Included in LWUIT are UI controls such as Labels, Buttons, RadioButtons, CheckBoxes, ComboBoxes, Lists and more.

Small Footprint API

Because of its small footprint, the LWUIT library can easily be bundled with an application, thereby making the deployment of LWUIT applications exceptionally easy. A JAR file size as little as 110kb is possible for a simple application. Larger file sizes are needed for richer, more complex applications, but are not much larger than those of typical MIDlet suites. Much of the footprint of a rich application is due to resources rather than class files.

Web Support

The HTML component introduced in LWUIT 1.4 enables applications to easily render HTML conforming to XHTML Mobile Profile 1.0. With this feature, developers can now include dynamic web-based content directly in their application, or leverage web-style development technologies including HTML and CSS to more easily design client-side mobile interfaces.

Themes

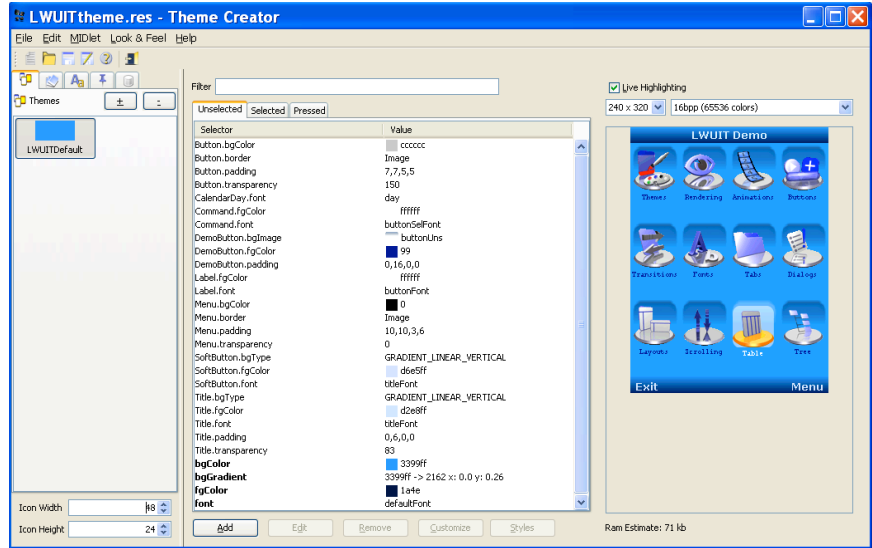
LWUIT supports a pluggable look and feel model. By applying different resource sets to an application, the theme or appearance of that application can be changed. Developers can easily create or modify themes by editing parameters with the theme creator.



LWUIT supports Theming and Pluggable Look and Feel (PLAF)

LWUIT Theme Creator

The LWUIT package includes a very powerful theme creator, which permits the rapid customization of applications implemented as a combination of Java code (MIDlet and LWUIT classes) and graphic assets (resources). The theme creator is intended for designers and decouples their environment from that of the programmer. With the MIDlet code loaded into the theme creator, the designer can easily manipulate resource definitions. Fonts, images and themes can be dynamically substituted, with the result immediately visible.



LWUIT Theme Creator

Key Features	
Key Features in the LWUIT Library	
Layout Managers	Contains all the logic for positioning LWUIT components. The following layout managers are included with the library: Border, Box, Flow, Grid, Group, Coordinate, and Table
UI Controls	The following controls are included in the toolkit: Label, button, text box, input field, list, table, calendar, tree, spinner, virtual keyboard, HTML component
Rich Text Support	The toolkit features bitmap fonts and bi-directional text support. Also included is a tool to create custom fonts
Animation	Motion (parabolic, spline, linear), Transitions (Slide, Fade)
Touch Screen	All LWUIT components support touch events. No special coding is needed to run LWUIT application on touch devices
Theming Support	Create a CSS-like file that can be loaded or changed at runtime, controlling the look and feel of the application
Mobile 3D Graphics (M3G) Integration	Enables the use of 3D effects (e.g. cube transition effect), drawing 3D elements within LWUIT applications, or using LWUIT drawn widgets within a 3D world
Web Support	HTML Component enables support for client-side rendering of dynamic web content, integration of existing web flows, and creation of HTML based user interface

Learn More

For more information about LWUIT, visit <http://java.sun.com/javame/technology/lwuit/>



Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

SOFTWARE. HARDWARE. COMPLETE.