Lightweight UI Toolkit

Release Notes

Release 1.5 for Windows

E24866-02

April 2012

Contents

Introduction

The Lightweight User Interface Toolkit (LWUIT) is a widget library that you can bundle with your application. Lightweight refers to the library's small footprint and moderate device requirements. With LWUIT, resource-constrained devices can have a rich look and feel. Inspired by Swing, the toolkit implements a peer-less graphics and UI model in a compact package.

LWUIT is an open source project. To join the community, visit `http://lwuit.java.net`.

Release Highlights

This release has the following highlights, as summarized in this section. For a list of major changes see "Major Changes in This Release", and for a full list of changes, see "All Changes in This Release".

- **GUI builder** - A tool for designing screen layouts by drag and drop Components on a palette, plus an IDE integration which allows developers to hook an action to the UI.
- **IO Framework** - Allows portable access to network, file system, storage, web services and a json parser.
- **New Ports** - Java SE port (allows testing, debugging, profiling and demonstrating the application on the desktop), and a CDC/PBP port for embedded devices.
- **HTML4 Tag Support** - Provides support for most of the HTML4 tags in the HTML component.

- **XML Parsing API**
- **New Components** - Tabs (replacing the old TabbedPane), Slider, Popup dialogs, and the Peer Component (allowing integration of native components "in place" within a LWUIT application) are new in this release.
- **Container list** - Allows for variable-sized lists, grid lists and similar effects.
- **Audio playback API** - Allows portable access to media.
- **Drag and drop API** - Supports touch devices and tablets.

Prerequisites

In general you will use the library with an IDE or a toolkit such as the Java ME SDK. If you have a recent version of the toolkit installed, its environment will support the addition of the LWUIT library.

A minimal environment is:

- Microsoft Windows XP or Windows Vista.
- Java Platform, Standard Edition Software Development Kit (Java SE SDK) version 1.6 update 4 or higher. To download the SDK software, go to `http://java.sun.com/javase/downloads/index.html`.
- Java ME SDK.

  The Java ME SDK is now a NetBeans plugin and included in the NetBeans "all" bundle in support of Java ME. Go to `http://www.netbeans.org` and download and install the latest full version of NetBeans.

Installing the LWUIT Binary

The downloadable LWUIT binary includes the toolkit libraries, the Resource Editor, documentation, and sample applications. This Toolkit is able to run on CLDC1.1, MIDP2.0/CDC PBP/SE, Java SE. The `lib` directory contains the following toolkit platforms:

- `LWUIT_MIDP.jar` - Standard MIDP/J2ME binary (works with standard MIDP 2.x/CLDC 1.1 devices)
- `MIDP_IO.jar` - LWUIT4IO's MIDP version, allows IO operations such as networking, storage and filesystem actions
- `LWUIT_Desktop.jar` - Works with Java SE applications, very useful for debugging/developing/demoing without a mobile environment
- `SE_IO.jar` - LWUIT4IO's Java SE version, allows IO operations such as networking, storage and filesystem actions
- `LWUIT_CDC_PBP_UI.jar` - Version of LWUIT compiled to run on top of the CDC PBP (Personal Basis Profile)

Using the Library With NetBeans

Follow these steps to add LWUIT to an existing project:

- Download and unzip the LWUIT binary.
- In the NetBeans Project Window, right-click on a project and select Properties. Choose the Libraries and Resources category and click the Add Jar/Zip button.
- Browse to select the appropriate platform from the `lib` directory, and then click Open. Click OK.

Using the Resource Editor

The Resource Editor has been redesigned and improved in many ways. In the `resource_editor` directory, double-click the `ResourceEdit.exe` file to launch the editor.

What's New In This Release

This section compiles the release changes. For a summary, see "Major Changes in This Release", and for a full list of changes, see "All Changes in This Release".

Major Changes in This Release

- Adapted the LWUIT Demo for tablet devices.
- Added a call to `hidenotify/shownotify` when `onDisplay/onUndisplay` occurs.
- Added a feature to install the default painter into a custom style (custom in the sense that it represents a state other than pressed, selected, unselected or disabled).
- Added a feature to the image border allowing a developer to specify arrow images and a destination component thus allowing a border for a component to point at another component. This is used by the `PopupDialog` to point at the component that popped it up if instructed to do so by the theme.
- Added a new animation mode to replace and wait where the layout is animated when replace and wait finishes.
- Added a new dialog transition type and made a slight improvement to the slide and fade transition.
- Added a new mode for the combo box where its popup can act as a spinner for platforms where it is uncommon to have a "real" combo box.
- Added a new popup dialog feature to the dialog demo.
- Added a new type of replace method that performs a callback when the replace completes and allows triggering a layout animation on completion rather than revalidation. This allows the replaced component to "slide into place" rather than jump into place.
- Added a properties mechanism allowing custom components to expose mutable state to the GUI builder. Added such properties to some of the complex components such as table, tree, tabs and HTML component.
- Added a simple 3D text effect style.

- Added ability for a component to indicate the "selected" rectangle which might be useful for many purposes (specifically for popup dialogs).
- Added ability to graphics to more easily get/set the clip as an array.
- Added ability to hide a component when in portrait mode, this is useful for UIs such as master detail where the master aspect is prominently displayed when room is available in landscape.
- Added ability to move the checkbox/radio button to the opposite side.
- Added ability to unify component groups for the toggle buttons in very large screens.
- Added ability to visually localize the VKB and the text field input modes.
- Added an orientation listener feature to Form.
- Added Audio playback API and support for built-in device sounds.
- Added built-in support for fading scrollbar and gradient fading edges of the screen.
- Added cache for the radial gradient backgrounds.
- Added experimental support for drawing scaled images and for an image border that scales the patches rather than tiles them.
- Added feature to disable the scaling feature of coordinate layout.
- Added `flushEDT` to the `callSeriallyAndWait`.
- Added Generic XML Parsing API.
- Added `GridLayout` support for dialog commands.
- Added high resolution images to the LWUIT Demo’s scroll demo.
- Added layout animations which are effectively an animated version of revalidate.
- Added new disabled style type.
- Added new drag and drop API.
- Added `PackTouchMenuBool` constant to allow the touch menu to size using the legacy size code.
- Added snap to grid functionality to component with supported implementations in container/list allowing the component to align according to a predefined grid when scrolled. This is an optional component turned off by default.
- Added support for a default icon in the tabs for some UIs that make no sense without an icon.
- Added support for declarative style inheritance, allowing style UIID reuse.
- Added support for fading image scrollbar in platforms where alpha transparency is supported.
- Added support for HTML generated content using CSS `content` and counter mechanism. This includes support for the following CSS properties: counter-reset, counter-increment content (including all of it's sub properties), before/after pseudo-classes.
- Added support for padding/margin based on millimeters (approximate) or screen size percentage, this is exposed in the style using the margin/padding unit when unspecified pixels are the default.

- Added support for platform APIs: `exitApplication`, `getProperty`, `execute`.
- Added support for tensile highlighting behavior for platforms that support that form of tactile feedback on overscrolling.
- Added support for the full range of CSS2.1 selectors (Above XHTML-MP1): Attribute selectors, Child selectors, Sibling/Adjacent selectors, First child selector, Language selectors.
- Added support for theme constant detection which allows for an undefined value, this enables the theme not to override an existing value with a made-up default.
- Added support for theme picking in the `UIBuilder` allowing us to view the UI with different themes far more easily.
- Added support for vertical aligning a text field/single line text area.
- Added tablet detection API.
- Added tablet support to the resource editor.
- Added the ability to customize default tab placement from the theme.
- Added the ability to define an overlay to the spinner.
- Added the ability to disable tickering by setting the ticker speed to -1.
- Added the ability to set a focused mode image for check boxes/radio buttons.
- Added the ability to tabs to switch on focus and the ability to customize the tabs container based on the tabs having focus.
- Allowed themes that have a title border to hide that border completely and also remove the preferred size inherited from this border.
- Applied the patch from thorsten in RFE 446: Fixed outer borders to work with image border and exposed the thickness property.
- Blocked the VKB from showing on none-editable text areas.
- `Border.java` accidentally exposed an internal implementation method.
- Changed the default themes of the LWUIT demo to use the tipster/timeline themes and removed the old themes, added ability to plugin external themes and use the native themes when available.
- Container list allows for variable sized lists, grid lists and similar effects.
- Demos now leverage the new project structure, feature versions for Java SE out of the box.
- Disabled the ends with 3 points behavior for touch commands which isn't quite right for touch buttons.
- Disabled width/height CSS properties since they cause in many cases some components to disappear or be very small.
- Fix for a table refresh error from the forum where table headers appear despite include headers being turned off.
- Fix for a tree replace issue reported in the forum that can be triggered when clicking a node multiple times fast.
- Fix for an issue when replacing a focusable component, this triggers a request focus which triggered a layout that broke the animation AFTER then replace.
- Fix for `bgImage` multi-image bug.

- Fix for issue 385 - Long Pressing * key on `TextField` causes Symbol dialog to appear repeatedly.
- Fix for issue 389 - Ability to disable virtual keyboard.
- Fix for issue 413 - `TextArea.getLines()` is incorrect when the last character is a newline character.
- Fix for issue 439 - added the option to drop events on `invokeAndBlock` and added an additional version of `replaceAndWait` that drop events.
- Fix for issue 464 from thorsten: lookup fonts don't fail gracefully when null is returned as a result.
- Fix for issue in transition when sliding between a scrolled down view.
- Fix for issue when replacing a theme to a theme that has a different menu behavior.
- Fix for issue where a component has `repaint()` invoked on it and is removed immediately after (this can happen as an edge case of replace even when used correctly), a dangling component instance might remain in the paint queue. It is now removed.
- Fix for issues 460 - Wrong size of combobox popup.
- Fix for touch bug where the a button action event code triggered by a touch event removed the button. The remove happened after the repaint entered the queue thus triggering the button to play at the top of the screen.
- Fixed a null pointer exception on text area that can occur if a click triggers a component removal.
- Fixed alignment of bottom aligned image.
- Fixed list pure touch selection to work with fisheye lists and to detect that use case (brought up by the tipster demo).
- Fixed padding in the table layout which didn't sum up the position of the cells correctly.
- Generalized the LWUIT demo to rely less on forms and work better for tablet form factors.
- Generic list renderer allows easier renderer creation based on container concepts, includes many features such as built-in tickering, check boxes, animated images etcetera.
- Hid a text field API blocking users from overriding and accidentally removing the code.
- Improved pixel density query to allow for none-square pixels.
- Improved the styling of dialogs to make more sense, this allows dialogs to be customized more effectively in the resource editor.
- Improved the title back command behavior to place 2 or 4 commands in the title bar area and added a new UIID for none-back title bar area commands.
- Improved the updating of the checkbox/radio button art when changing a theme by clearing out the variable values.
- Incorporated the virtual keyboard into the main implementation and deprecated the VKBImplementation. Improved the integration with the native Virtual Keyboards when applicable.

- Issue 457 `Display.shouldRenderSelection()`: Rather complex patch adapting `shouldRenderSelection` to work in a case where a repaint can be triggered on an unrelated component.
- Issue 458 `Component.isDragActivated()` behavior: added a global drag activated flag which solves the issue with none-scrollable parent drag.
- Java SE port allows testing, debugging, profiling and demoing the application on the desktop or within a browser Applet.
- Lead component feature, allows complex container hierarchies to act as a simple component and delegate their event handling logic to a single "lead" This allows for easy construction of elaborate UIs such as multiline buttons without much code and without inheritance.
- LWUIT4IO allows access to filesystem, storage and networking in a cross-platform way.
- Made the cancel command of the combo box map to the back button of the device when applicable.
- Minor fixes for the new popup demo and theme fixes so the UI will look decent in the popup demo.
- Minor improvement for tab behavior when focused.
- Minor performance improvement for box layout.
- Moved allot of the LWUIT Demo resources into the `src` directory and added larger multi-image versions of the icons of the application suitable for high resolution devices.
- Moved the initial pointer pressed members to the Form.
- Moved theme constants to the look and feel so we don't need to perform a constant lookup for every constructed component.
- Multi-Image and Improved SVG support allow targeting multiple resolutions more easily by providing resolution specific graphics.
- New animation demo in the LWUIT Demo to better match what we think of today as "animations".
- New command behavior functionality allows defining how commands are added to LWUIT. Allows mapping commands to native menus, touch menus, back button in title, menu key or softbuttons.
- New GUI builder API the UIBuilder.
- New `LayeredLayout` layout manager allows stacking elements one on top of another.
- New peer component support allowing integration of native components "in place" within LWUIT applications.
- New project structure allowing greater portability of applications and removes reliance on MIDP.
- New slider component, provides both vertical and horizontal progress indication as well as an editable mode where the user can modify the slider to indicate things such as volume.
- New Tabs component replacing TabbedPane allowing side swipe gestures to switch between tabs and adding greater ability to style/customize the tabs.

- New Theme constant functionality allowing customization of various LWUIT properties such as transitions and dialog positioning from the theme.
- New Tipster demo application.
- New `VideoComponent` simplifies the cross platform interaction with videos.
- Performance improvement for very large numbers of components specifically in calculation in the Container class that seemed to slow down considerably for HUGE numbers of components.
- Placed tab text bellow the icon by default which is the most common configuration.
- Prevented an edge case null pointer exception in text area.
- Prevented potential division by 0 mentioned by thorsten in issue 398.
- Removed old deprecated code and considerable API cleanup.
- Removed the old drag and drop code in the LWUIT demo and replaced it with the generic LWUIT drag and drop support.
- Renamed the Tabbed Pane demo to the Tabs demo.
- Reset the calculated preferred size on `setUIID` which might affect the preferred size.
- RFE 455 Copy and Paste API: Added support for a simple copy and paste API which by default just stores everything in a LWUIT clipboard but can/should map to a native clipboard when available.
- RFE 456 Feature request: device orientation. Added support for requesting/querying device orientation (haven't changed LWUIT to use it though). This is only implemented as a generic stub.
- Template support in the resource editor.
- Toggle button support allows checkboxes/radio buttons to be rendered as toggle buttons.

All Changes in This Release

This is a detailed list of all changed in Release 1.5.

- Added a "name" attribute to components.
- Added a client property feature to commands.
- Added a command behavior API that allows changing the way commands are processed by LWUIT, this replaces and unifies all the different ways to customize the command menu (touch commands, native commands etc.).
- Added a constructor to `PainterChain`.
- Added a default document request handler which works with JAR local resources and res files.
- Added a default value for `List.fireOnClick`.
- Added a DPI density API in for the multi-image feature.
- Added a feature allowing menus to assume their preferred size on the screen.
- Added a feature to border layout allowing it to absolutely center an entry within the center constraint rather than stretch that entry.

- Added a feature to `FlowLayout` allowing it to grow to accommodate an entire row, this is very useful for the touch menus which now also allow such a flow layout as an option.
- Added a feature to grid layout allowing it to occupy an entire row even when the number of columns doesn't divide perfectly with the number of cells. This feature is integrated into the touch menu to make it more appealing.
- Added a feature to `GridLayout` allowing it to automatically calculate the optimum number of columns/rows based on content preferred size.
- Added a feature to keep tensile drag always on.
- Added a feature to prevent the virtual keyboard from folding automatically on form switching.
- Added a grid list feature to the scroll demo to show off the `ContainerList` class.
- Added a more robust `readAll` method that works similar to `readFully` but doesn't expect a known array size.
- Added a resource editor ability to extract localization data from the UI builder resources.
- Added a resource editor dialog to list image sizes in the resource which allows us to see why a resource of size X is taking so much storage.
- Added a separate border mode to table (A border surrounds each cell) - default is the previous one (collapsed).
- Added a title area container that allows placing additional elements in the title (at your own risk), this is used internally for features like a title based back button.
- Added ability to "lock" component painting useful for transitions, this allows locking the painting of components so they draw once into a buffered image (optionally weak reference) and then paint only that image.
- Added ability to access the listeners of the list to allow seamlessly transferring listeners from a List to a `ContainerList`.
- Added ability to define disabled state checkbox/radio button images.
- Added ability to disable tickering globally in LWUIT.
- Added ability to set a property in display, this is mostly useful for cases where applications have business logic that needs to override application wide defaults.
- Added accessibility theme templates to the resource editor.
- Added additional tile abilities allowing to tile background images on the top/bottom/left/right and center of a component.
- Added alignment/text decoration support for resource files.
- Added an ability to "preposition" a specific dialog in the screen, this allows us to give dialog X a coordinate in advance of showing.
- Added an easier way to associate a radio button with a specific group.
- Added an empty background option for style to allow a UIID deriving from another style to override background image/gradient settings with blank.
- Added an event framework to `HTMLComponent` that allows dispatching events from controls (textfield, buttons etc.) via `HTMLCallback` and to alter the DOM dynamically according to user input in such controls. Use

setEventsEnabled(true) to enable - it is disabled by default to save resources.

- Added an HTML demo to the UI demo application.
- Added an init() callback to the implementation that's invoked on the EDT right when it starts. This allows an implementation that depends on binding logic to a specific thread to make use of that feature.
- Added an optimization for the use case of repeatedly loading the same resource file which might happen in the UI builder. Also added the ability to cache the resource file in the UIBuilder from now on.
- Added an option for an outer border (a border of the border).
- Added an option to disable auto focusing on first link in HTMLComponent.
- Added background none to the style options in the theme editing.
- Added benchmarking tool to allow measuring the cost of a theme/UI in proportion.
- Added call for repaint when removing a painter from a paintchain as setGlassPane is not called in that case, but still the painterchain changes.
- Added cell spacing for tables (increases margins between cells) - both horizontally and vertically.
- Added embedded container support to the UI builder which allow embedding a UI resource into a form.
- Added ExtHTMLComponent which is an extended HTMLComponent that has a pointer mode, which can be helpful on some non-touch devices (Similar to other mobile browsers) This mode can be toggled using the ExtHTMLComponent.setPointerEnabled() method. Added shortkey 8 to toggle this mode, and added a menu item for it.
- Added feature to allow tabs to grab up all available space on the screen (horizontally) as a side feature they can now also break lines when running out of space.
- Added feature to resource editor allowing the import of a separate resource file.
- Added focus listeners to Tabs Component and removed the final from the class definition.
- Added getElementById returning an element with the given ID (in the sense of the ID attribute).
- Added grabsPointerEvents to resolve issue 430: getComponentAt(xy), containers and z-layout.
- Added help information to multiple dialogs within the resource editor.
- Added horizontal/vertical image borders that can only grow on one axis but provide very accurate asymmetric UI replication.
- Added HTML4 tags, mainly: Image maps (MAP, AREA), Subscript and superscript (SUB, SUP), Legend for fieldsets (LEGEND).
- Added image/border locking feature which optimizes the EncodedImage objects to prevent us from constantly recreating/discarding them in image heavy UIs by notifying the image when its showing on the screen.

- Added information to images in the resource editor, specifically showing the components in the UI builder/theme making use of a specific image (or another timeline image). Also in a case of an image that is a part of a border we now show a preview of the border.
- Added information to the action event allowing an event broadcast by a button with a command to contain both the button and the command.
- Added inner borders modes (draw all, columns, rows, groups).
- Added `isInitialized` method to Display.
- Added line wrapping to the i18n editor of the resource editor.
- Added links to help videos to the resource editor.
- Added more elaborate border layout configuration features allowing the center alignment of the component to be more refined (either center by available space or absolutely), this was needed by the back button title bar to create proper centering of the text.
- Added new cell renderer interface which should supersede the `ListCellRenderer` in the future.
- Added preview on device feature for the resource editor. .
- Added readable conventions for the Display.getProperty method to allow cross platform lookup for common lookups (e.g. app name/version) that should work regardless of the platform.
- Added resource editor video tutorial entries into the help menu allowing designers/developers easier access to information about the tool. This way the help is video-based and live with the ability to include comments/discussion.
- Added `setEnabled` to Container, which will enable/disable all Container's children.
- Added support for 11 common HTML4 tags that are outside the XHTML-MP 1.0 spec (including U for underline).
- Added support for an enabled flag on commands.
- Added support for CDATA sections in XML - they will be added as text nodes to the document.
- Added support for commands to include rollover, pressed and disabled icons so they can be reflected in menu.
- Added support for customizing the VKB and changing the text field input modes from a localization file with visual editing support.
- Added support for device density (DPI) which is currently mapped/defined as a set of resolutions but ports to more advanced platforms can map this to more sensible values.
- Added support for exiting the `RunnableWrapper` loop when exiting LWUIT.
- Added support for flattening a component thus allowing it to paint the layered background in a more efficient yet memory intensive way.
- Added support for image scaling to the image download service.
- Added support for obtaining the underlying native graphics object during paint methods which will allow integration with elaborate native APIs that require the graphics object context. This is achieved by storing the state of the LWUIT

graphics, before returning the native graphics then restoring the original state when the graphics is returned. It is still recommended to return the graphics object to its original state.

- Added support for password protecting resource files.
- Added support for placing a hint on the list that will be rendered when the list contains no elements similarly to the feature already available in text area.
- Added support for SoftReferences on platforms that support it.
- Added support for the `cellspacing` and `cellpadding` attributes in the TABLE tag.
- Added support for the `direction` CSS property.
- Added support for the following CSS properties: `"border-collapse", "empty-cells", "border-spacing", "caption-side", "word-spacing", "line-height", "min-width", "max-width", "min-height", "max-height","quotes", "outline-width", "outline-style", "outline-color"`
- Added support for the `noframes` and `noscript` html tags.
- Added support for the rules and frame attributes in the TABLE tag (allows drawing partial segments of the inner or outer borders).
- Added support for the TBODY, THEAD and TFOOT HTML tags.
- Added support for the text-decoration CSS property.
- Added the ability for a border to declare a minimum size thus enforcing the UI to use said size at a minimum. This is implemented for the image border which now makes sure not to shrink bellow its corner sizes which allows things like gradients to always look good. Our previous solution of requiring padding to workaround this issue wasn't practical since font sizes change between platforms making padding inappropriate.
- Added the ability for the locale to determine RTL orientation and for the resource editor to enforce a specific locale.
- Added the ability to add multiple SVG images in the resource editor.
- Added the ability to `callSeriallyAndWait` with a timeout to prevent a potential deadlock. Normally no one should use that method but its essential for some edge cases where you explicitly know you are violating the EDT and still want to do that which is a case with the resource editor which must either violate the LWUIT EDT or the Swing EDT and runs into exactly that problem.
- Added the ability to get an event when a form is shown.
- Added the ability to globally determine the default list behavior (cyclic/none etc.) from the theme.
- Added the `AsyncDocumentRequestHandler` interface which allows providing async implementations.
- Added the border attribute to the HTML img tag.
- Added the disabled and readonly properties and applied to all relevant HTML tags.
- Added the following border styles: Double, Dotted, Dashed, Groove, Ridge, Inset, Outset.

- Added the full table of HTML color names (with matching values).
- Added the HTML ismap property to the img tag, allowing server image maps.
- Added the overline text decoration to Style.
- Added the segment concept, allowing defining logical table segments (currently used for INNER_BORDER_GROUPS, but it's in the model and can be used for other issues).
- Added timeline transition feature allowing timelines to represent the mask image for a transition.
- Added to border layout the ability to rearrange itself based on the devices portrait/landscape mode. When this feature is active all border layout activity is assumed to be targeted at portrait devices, when landscape mode is activated in the device (device width > device height) the layout changes based on the defined constraints. This allows shifting of logical UI component positions based on device orientation.
- Added to the implementation the ability to exit from an application if supported by the platform.
- Added toString and comparability to layout managers.
- Added vertical alignment support to flow layout.
- Alignment of components was moved from Label/TextArea to the style class which means all the old alignment code is now deprecated and some odd things might happen because of selected/unselected/pressed alignments differing one from the other. Please let us know if you see such issues and we will try to help.
- Allowed bidi sensitive constants to refresh when a theme is updated.
- Allowed implementations to cleanup after themselves.
- Allowed zooming images in the resource editor to see them more clearly in the UI.
- Also provided the List with an ability to calculate its preferred size based on a larger pool of cells (although this is quite inefficient).
- Attempted Fix for issue 377: NumberFormatException in LookAndFeel.refreshTheme().
- Border - allowed compound borders to have empty sides.
- Bug fix when negative margin/padding exists they are ignored (so not to cause an exception).
- Button now fires events on pointer release only if it was pressed first which prevents an issue where users drag their finger into a button and release by mistake which can happen on touch device UIs that are comprised heavily of buttons.
- Changed names of public methods in Element and HTMLElement: all get..byName to byTagName and get..byId to byTagId, due to the fact that both name and id have double meanings in XML/HTML.
- Changed pure touch mode to no longer render selected entries as selected when no selection is rendered behind the list.
- Changed the day localization keys to comply with the Month localization keys.
- Changed the default behavior of the list to not render the focus component when it doesn't have focus.

- Combo box popups now have a long requested overhaul for their popup logic allowing overriding the placement of the popup dialog both by the theme and programmatically. We also allow for different renderers between the combo and the popup which was also requested in the forum. This is activated via a constant flag.
- Complete resource editor rewrite of the UI for the theme editing with a "simplified" list of properties and streamlined properties editing dialog.
- Compound borders are done now via the code in Border which was moved from CSSBorder (the latter is deleted in this version).
- Deprecated `setListCellRenderer()` so `get/setRenderer()` will confirm to the JavaBeans standard.
- Deprecated the Log class, a somewhat compatible class is now in LWUIT4IO. This class relied on APIs that aren't as portable.
- Dialogs now have their own content pane which is NOT the standard Form content pane and their own title (again NOT the form title). The reasoning for this is to solve a serious styling of dialogs hack that required complex treatment of the dialog/dialog title in a way that didn't make sense. One can now define a Dialog border in the resource editor and actually get something that works as expected with/without the title.
- DOM manipulation is supported via the Element/HTMLElement classes.
- Eliminated the `bgAlign` property effectively and replaced it with more entries to the `bgType` making the UI/programming of background alignment much simpler.
- Enabled SVG preview in the resource editor preview file open dialog.
- Enhanced encoded image to allow for some unique customizations including lazy loading from file system etcetera.
- Exposed a setter to the hint icon and fixed the hint text method in text area to allow the GUI builder to better customize them.
- Exposed alignment in flow layout and some minor features for the resource editor.
- Exposed dispose when pointer is out of bounds in the dialog so the UI builder can make use of that.
- Exposed the ability to place commands as buttons within a dialog, this feature was used by the dialogs static show methods but was internal in the Dialog class. It is now public and also includes some more elaborate styling.
- Exposed the native font for some elaborate implementation and platform specific uses.
- Exposed the show native screen feature from the LWUIT implementation.
- Exposed the text decoration feature of the style properly and fixed some inconsistencies within it.
- Exposed `validChar` and fixed `VirtualKeyboard` to valid the chars inserted according to the given `TextField`.
- Fix for a border layout bug which always returned WEST for a component that has no constraint. This can happen when changing the layout AFTER adding the components (common in the UI builder).

- Fix for a constantly recurring regression with fast scrolling in list while holding the down/up button, it seems animation was de-registered too soon due to a bug that effectively ignored the key if an animation was in progress. This now changes the appearance of fast scrolling but for a change it actually works and doesn't result with focus hanging outside the list boundary. Previous "fixes" for this issue triggered a bug where animations were never de-registered after the fact effectively seeping CPU.
- Fix for a couple of tab bugs one of which only relevant to the resource editor and another style issue only relevant to the resource editor.
- Fix for a major bug in image rotation behavior .
- Fix for a transition issue when transitioning between two components that are within a scrollable container they can be much larger than the screen size and cause an exception.
- Fix for a unique case of exception reported in the forum in the `DefaultLookAndFeel` that can occur in CJK languages rendered in a label in a list cell renderer.
- Fix for an issue where we layer themes one on top of the other the constants of the second theme weren't processed since `refreshTheme` wasn't invoked again.
- Fix for an issue with back navigation in some edge cases of transitional forms.
- Fix for an issue with List selection being outside of the scroll viewport for some special cases on some devices. I'm not exactly sure why that happened (probably a race condition due to a thread bug that I missed) but this works around that issue.
- Fix for bug 354 - TextArea and form scrolling/height problem.
- Fix for implicit issue mentioned in 429: `http://java.net/jira/browse/LWUIT-429.` When clicking a softbutton in a dialog that has `commandsAsButtons` active the physical softbutton keys don't work as expected.
- Fix for installing the button menu bar in devices that don't have softkeys which is naturally a large demographic for this feature.
- Fix for issue 329: `refreshTheme()` doesn't honor custom BG painters.
- Fix for issue 332: `ArithmeticException` thrown when using `scaledWidth` or `scaledHeight`.
- Fix for issue 360: invoking `repaint(int,int,int,int)` might interfere with an existing `repaint()` (no args) call by setting the `dirtyRegion` field to a partial value.
- Fix for issue 363: Typo in `form.java` size change handling.
- Fix for issue 376: `LabelForComponent` is not working for TextField, working for `TextArea` in LWUIT 1.4.
- Fix for issue 423 from Thorsten: Method returning system font sizes incorrectly returned pixel size.
- Fix for issue 425 simplifying the process of inheriting the border class.
- Fix for issue 431: regression due to lead component patch.
- Fix for issue 440: `Motion.getFriction()` - edge case issue.

- Fix for issue 454 from thorsten related to minor issues with scroll fading implementation and a couple of tensile drag improvements.
- Fix for issue in adding a layout component with an index.
- Fix for issue in Button which didn't override its `setUIID` method to nullify the pressed state.
- Fix for issue with title containers that prevented them from disappearing when there was no content in the title but the title had an image border. This was triggered by the minimum size for an image border feature.
- Fix for list behavior in pure touch mode which doesn't seem to be correct.
- Fix for long existing bug in list renderer size calculations that didn't take container caching the preferred size into calculation when running the algorithm of sizing based on the first 5 entries. The fix for this required exposing the `setShouldCalcPreferredSize` method to allow access from the look and feel calculation code.
- Fix for null pointer exception in the `GameCanvasImplementation`.
- Fix for radial gradient positioning when their diameter is different than 1.
- Fix for some menu bar issues specifically in touch menu detection and the usage of the global commands rather than the commands passed to the method in the case of the touch menu.
- Fix for synchronization issue in the log requested in the mailing list.
- Fix for text area out of memory error on infinite loop.
- Fix for text wrapping in languages that need text width size by string and not by char (e.g. Arabic).
- Fix for the virtual keyboard moving the anonymous inner class to a static inner class thus removing the reliance on the behavior of the parent class field key and access to the parent class. This caused problems in some unique circumstances which were hard for people to figure out (justifiably) since the behavior relied on rather esoteric behavior of the JVM generated code when creating inner classes and allowing them to access parent classes.
- Fix for the width and height CSS properties (when used before the size of the component would be 0).
- Fix for tickering issue in the generic list cell renderer which always tickered based on the first entry since the selection component is always created based on the first entry.
- Fixed a bug in button where one of the alternative icons (pressed/rollover/disabled) didn't get animated.
- Fixed a bug in `ResourceThreadQueue` that caused NPE when no HTMLCallback was set and an image was not found.
- Fixed a bug in the title button bar that caused tickering if a title with short text was replaced with a title with long text, this happened in the LWUIT demo when entering the HTML demo.
- Fixed a few issues in Form - the `focusScrolling` initialization was inside an if statement by mistake.
- Fixed a form resize issue that prevented a hidden form from updating its size causing an unattractive transition if a rotation of the screen occurred in between.

- Fixed a performance issue with TextField drawing.
- Fixed a regression in Table cell editing and virtualKeyboard.
- Fixed a VERY old LWUIT bug causing the dialog to resize itself on `refreshTheme`.
- Fixed addition of a painter to a painterchain - the previous code called `addPainter` and did nothing with the return value which is the new `PainterChain` that needs to be set.
- Fixed alignment in button to also support pressed state alignment in the style (pressing the button returned to the default alignment).
- Fixed all of the resource editors image combo boxes to use a single code base that supports both a larger icon preview and keyboard searching.
- Fixed an arithmetic exception with an empty grid layout for an auto fitting grid.
- Fixed an emulator issue relevant for Torch device.
- Fixed an incorrect vkb lookup bug.
- Fixed `Border.equals` so it now compares correctly compound borders (This affects for example HTML elements that received several partial borders from different CSS classes, and needed to merge into one compound border).
- Fixed bug 390 (NPE possible in `HTMLComponent.displayPage`).
- Fixed bug in border color in HTMLComponent (when not specified - take theme colors).
- Fixed bug in descendant CSS selectors.
- Fixed bug that resulted in wrong rendering of tables with padding.
- Fixed bug that some pages got stuck after stopping/canceling.
- Fixed cases of HTMLComponent not showing the entire page (by revalidating the parent form if it exists).
- Fixed compound border so that if all 4 sides are the same - it creates a regular BORDER.
- Fixed CSS borders issue on multiple words.
- Fixed encoding bug when setting an HTML using `setTextBody`/HTML.
- Fixed focus scrolling to set the default in the constructor otherwise its impossible to apply it using the standard form setter method.
- Fixed font size recalculation on CSS font change.
- Fixed form reload in the UIBuilder to disable all the transitions.
- Fixed HTML lists indentation.
- Fixed `HTMLElement` to parse names attributes.
- Fixed issue when multiple roots are present at an XML/HTML.
- Fixed issue with excessive spacing.
- Fixed issue with zero length attributes.
- Fixed key events to work more consistently in the resource editor.
- Fixed MANY SVG bugs, SVG images now work as expected and on devices.

- Fixed mmapi nullpointer exception.
- Fixed numeric char entity rendering.
- Fixed ordered list numbers alignment in the HTML component.
- Fixed pure touch behavior to hide selection while dragging, this includes a fix in button as well. Exposed the pure touch implementation flag which is useful to block tickering in pureTouch which are usually disabled by default (since there is no focus).
- Fixed race conditions in the HTMLComponent that caused it to fail on the resource editors GUI builder.
- Fixed `refreshTheme` to revalidate the Form.
- Fixed spacing between words when white spaces are involved - for example o`ne <b>two</b> three` - would render before as `one twothree`, since leading spaces were omitted automatically. Now we consider the word position in the paragraph and add a leading space if needed.
- Fixed style bug with links press (font now stays the same size).
- Fixed text field/area preferred width to add the padding before the image border minimum size is used.
- Fixed the background image aligned top style to center align.
- Fixed the fade transition on global alpha true platforms.
- Fixed the flickering in the resource editor.
- Fixed the generic list cell renderer to restore focus to the list.
- Fixed the hint preferred size so the list/text components take up enough space for the hint label.
- Fixed the look of the back button title bar command behavior.
- Fixed timeline animation scaling.
- Fixed unordered list indentation to have a bigger indent in level 2 and beyond in the HTML Component.
- Fixed white spaces between tags.
- Generic renderer now sets the focus only to focusable subcomponents thus allowing more powerful styling for its child components.
- `HTMLComponent` autodetects if the implementation is `sync` or `async` and activates the appropriate methods.
- Improved error detection in image creation since some platforms just return a null image instead of throwing an exception.
- Improved lots of setters/getters to make the task of the GUI builder reflection more intuitive.
- Improved the performance of `replaceAndWait` using locks.
- Improved the text area hint API.
- Improved the UIID list in the combo boxes for UIIDs to include all the UIIDs declared in the theme.
- Improvements for list animation which again doesn't de-register itself when animation finishes.

- Initial work for improved native platform integration including new APIs for mixing a LWUIT based UI with a platform native component and interacting with said component.
- Initial work on a resource editor effect menu, specifically introducing the pulsate effect.
- Line borders can now have titles which are displayed on the top line (Top padding should be applied).
- LWUIT-wide Underline and Strike-Through support (see `Style.get/setTextDecoration` and `get/setUnderline`, `get/setStrikethru`).
- Made bidi in the implementation class static so deinitialize/re-init don't lose the bidirectional state.
- Made button press/release methods public.
- Made resource editor images use checkered background to make it easier to distinguish some image types.
- Made `showMenu` public allowing external code to trigger menu display with custom buttons.
- Major fix for opaque aligned image which failed on a LWUIT optimization that seamlessly sets backgrounds to opaque, however if the image is aligned it might be seen.
- Major fixes for background aligned image painting issues, also added 4 additional alignment destinations (top left, top right, bottom left, bottom right).
- Major re-architecture of the native integration to support the new APIs committed into LWUIT just now. There are still some issues related to the camera capture bug in the issue tracker which I hope to resolve somehow but generally the API seems to work for basic use cases. Still need to improve touch integration with native components.
- Many fixes to the table layout allowing it to reflow and work in a more robust way when mutated.
- MenuBar - added the softbutton area only on devices with enough soft keys.
- Minor additions to list allowing the theme to define the default list item gap which for historical reasons is 2 by default. Also allowing the theme to define the renderer behavior when the list doesn't have focus.
- Minor fix to the flowlayout resolves some HTML issues where tables and flowlayouts were mixed.
- Mixing of heavyweight (platform specific components) and lightweight (LWUIT components) is potentially problematic and might cause some basic LWUIT features such as z-ordering/glasspane to fail for such cases. In features such as dialogs the underlying native component might disappear when the dialog is shown but should work none the less.
- Moved `isEnable` use to the Form, rather then keep it in the Buttons level.
- Moved resource editor image menu functionality into the Image menu.
- Moved the button pressed style up into component to enable lead component to work more effectively with pressed buttons.

- Moved the grow animation from its hidden location in "Effects" to the Component class.
- Multi images are now supported both for rasters and SVGs providing far more resolution independence.
- Multiple bug fixes for the tabs component which had some issues such as a race condition when selecting a tab programmatically. Layout size issues preventing the Tabs from being placed in a flow layout.
- Optimizations for cases in which no external/embedded CSS segments exist.
- Parser and DOM APIs are now exposed and can be used/extended.
- Performance and memory improvement: All words in a text segment get now the same Style object, thus saves memory and also boosts performance when CSS is involved.
- Performance improvement for timeline animation, needs some more work.
- Potential workaround for GTK exception in font editor.
- Removed calls for `contains()` instead of `containsKey()` on Hashtables which effectively meant all hashtables that used this method didn't work.
- Removed MANY softbutton hacks that existed mostly for the purpose of overriding the style of the softbuttons in a particular way. This essentially opens up the softbuttons to "proper" customization via styling in LWUIT.
- Removed the deprecated border painted method.
- Significant performance improvement for HTMLComponent due to a change in event dispatcher.
- Resource Editor now supports keyboard navigation in the UI preview, arbitrary preview size/font size, defining device types in the UI preview (touch/none-touch, 1 menu button etc.) and overriding localization by the UI preview.
- Simplified styling handling in Calendar.
- Small bug fix, title events are now dispatched as usual (In the last revision there was a problem).
- Small improvements to the RSS reader allowing users to customize the destination form in the list.
- Some generic HTML utilities moved and exposed at HTMLUtils.
- Style events are not dispatched when in non-EDT thread unless a special static flag is set in `EventDispatcher`.
- Tuned the behavior of transitions in the theme to match what users expect more closely.
- XML Package was added, can be used to parse XML documents of all types.

Documentation Accessibility

This release includes PDF documentation in the binary download. For an HTML version of the documentation that is suitable for screen readers, please visit the Java ME Developer Tools documentation page at http://download.oracle.com/javame/developer.html and choose LWUIT 1.5.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/us/corporate/accessibility/index.html.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Lightweight UI Toolkit, Release 1.5 for Windows
E24866-02

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

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. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.