Oracle FLEXCUBE Direct Banking

Mobile Client User Interface Guide
Release 12.0.3.0.0

Part No. E52543-01

April 2014

ORACLE®

Mobile Banking User Interface Guide
April 2014

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India
Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax:+91 22 6718 3001
www.oracle.com/financialservices/

Contents

Preface

Intended Audience

The audiences for this document are

- IT teams of the Banks or Financial Institutions implementing the Oracle FLEXCUBE Direct Banking solution
- Implementation Partners who shall rollout the product for Banks or Financial Institutions
- Any other external party who shall be involved in understanding, extending, developing the User Interface of Oracle FLEXCUBE Direct Banking

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to OFSS Support

https://support.us.oracle.com

Structure

The document provides introduction to various UI components used in the browser based mobile banking and Java application based mobile banking.

Mobile web browser which is used in the mobile phones and recognized by many names like microbrowser, minibrowser. The main aim of the mobile browser is to display the web content efficiently on the mobile screens which are naturally very small in size comparatively to normal computer screen. Mobile browser should be small but able to hold the low memory capacity & low bandwidth of wireless devices. Maximum modern mobile phones provide inbuilt web browser having capabilities of HTML, CSS, & WML. Today in modern times mobile web access is powered by various big multinational companies which are producing efficient mobile web browsers, plus multi-platform browsers such as Opera Mini, Safari, Microsoft IE for Mobile, and many more.

Assumptions

The document assumes the audience has a good understanding of web technologies like HTML, Java Script and Cascading Style Sheets etc.

Also, the audience is expected to have a basic understanding of the server side technologies used within the application to render the user interface like XML, XSL, XSLT etc.

There are several technologies which are used by the web browser developers to develop mobile web browsers and making capable for the connectivity with internet.

Introduction

Utilization of the web via mobiles is on the rise! Accessing the internet through mobile phones has become necessity of professionals as well as general mobile users in this modern world of communication.

Oracle FLEXCUBE Direct Banking is a multi channel e-banking platform with support for customer touch points like Internet, Mobile Phones. Oracle FLEXCUBE Direct Banking is based on the industry standard Java Standard Edition (Java SE) and Oracle Java Enterprise Edition (Java EE) platforms.

Browser Based Mobile Banking - User Interface Design

XHTML Standard

The user interface conforms to the XHTML 1.1 standard.

The XHTLM 1.1 standard allows the following DTDs to be used within the pages.

```
<!DOCTYPE html

     PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">



<!DOCTYPE html

     PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">



<!DOCTYPE html

     PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"

     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Oracle FLEXCUBE Direct Banking has adopted the Strict DTD of the XHTML 1.1 standard for support of the internet banking user interface. All pages are automatically embedded with the DOCTYPE when the page is rendered using the XSL transformation techniques.

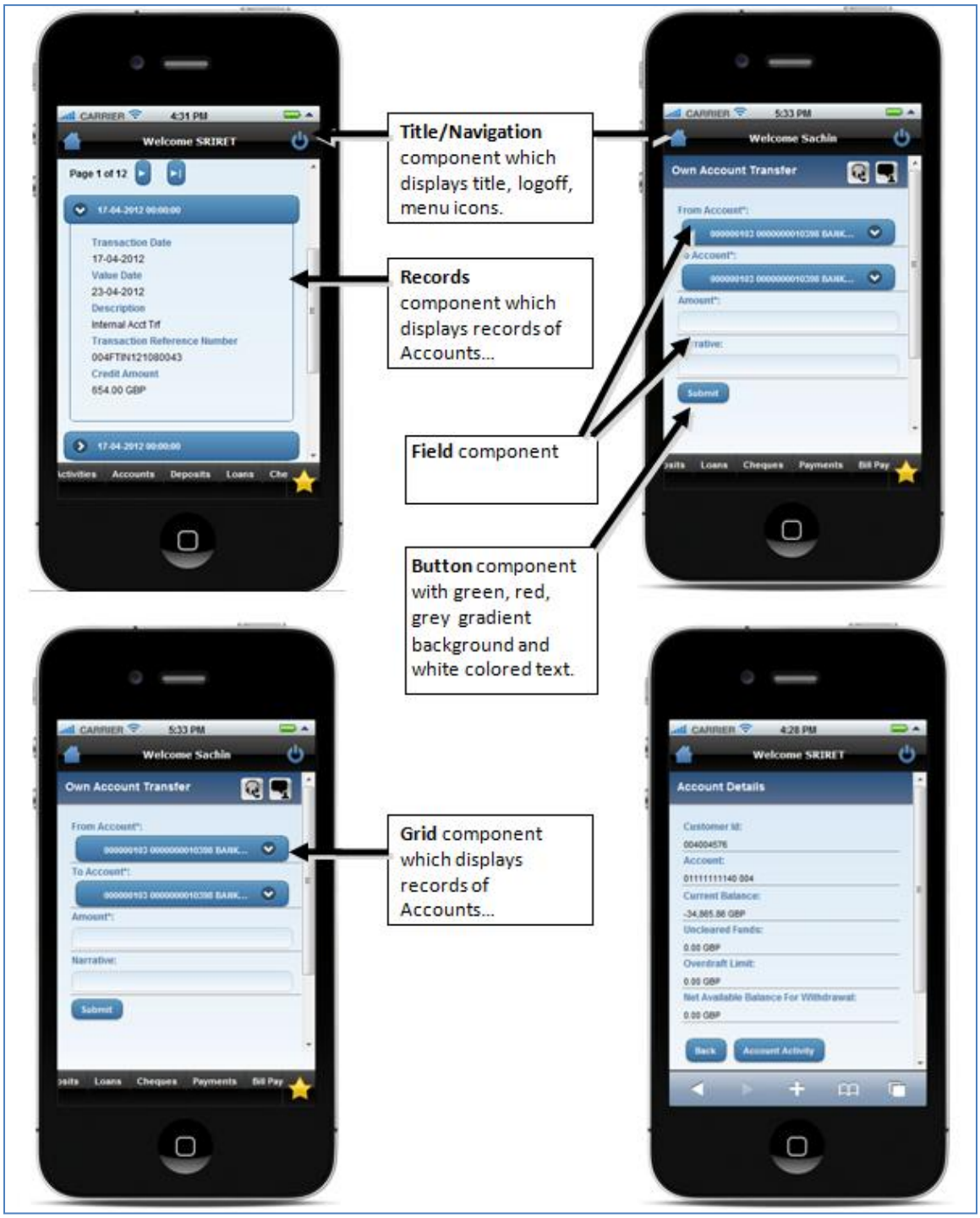
Viewport Metadata

iPhone web apps support the viewport metadata to give information about the zoom and content area that you want to use in your website. The browser for Nokia devices based on Symbian OS automatically adjusts the content to the screen, so there is no support for this feature.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"></meta>
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0,
minimum-scale=1.0, maximum-scale=1.0" ></meta>
<meta name="apple-mobile-web-app-capable" content="yes" ></meta>
<meta name="apple-mobile-web-app-status-bar-style" content="default" ></meta>
<meta name="HandheldFriendly" content="true"></meta>
<meta name="MobileOptimized" content="width"></meta>
```

iPhone/Android Components

These scripts are typically hard-wired to the particular page; if the content of the page changes, the scripts may have to be rewritten, since there is no way of abstracting one from the other. HTML Components provide this ability. Following are some examples of components.



iPhone/Android Login Page

Below is a screen shot of the Login page. It consists of the logo which is placed on the top center, a user ID and password text fields, a sign in button.

1) **Oracle Flexcube Logo:** The Logo is placed in a div at the top left corner with class name for logo is “oracle_logo_login” and title text class name is“logotext”.

2) **Login Section:** The Login section is placed in a table with class name is ”MainLoginTable”. Text input fields for User ID and Password is placed on the left .

iPhone/Android Landing Screen

Once the user has logged in, the FLEXCUBE application menu appears.

1) **Home :** On click on Home icon, will return to login screen.
2) **Welcome Details :** Welcome user details placing on the top bar.
3) **Log off :** Logout from current screen.
4) **Total Positions:** Shows list of various account types with balance details.
5) **Account Type Details :** Shows list of account details
6) **Back :** Return back to Total Position screen
7) **Menu:** Shows menu which scrolls horizontally.
8) **Favorites:** Shows selected menu as favorite.

iPhone/Android Components – IFrame, Input, Combobox, Error Message

After logging in, transactionpanel div tag displays iframe (1) in which main transaction screen appears with Combobox (2), Input Text Field (3), buttons (4), and error message dialog (Collapse/Expand) (5).

Cascading Style Sheets

Style Sheets of the FLEXCUBE application are available in the “css” folder. Only one styles sheets are maintained for both the Login page and the Transaction pages.

- Login Page and Transaction Pages : eng_42.css

In the Login and Transaction page, the CSS is called in the head section as given below, when user agent set to iphone, blackberry, android or nokia, path for CSS also change resp.

For iPhone

```
<link REL="stylesheet" TYPE="text/css" HREF="css/iphone/ THMDEFMOB/eng_42.css "></link>
```

For Android

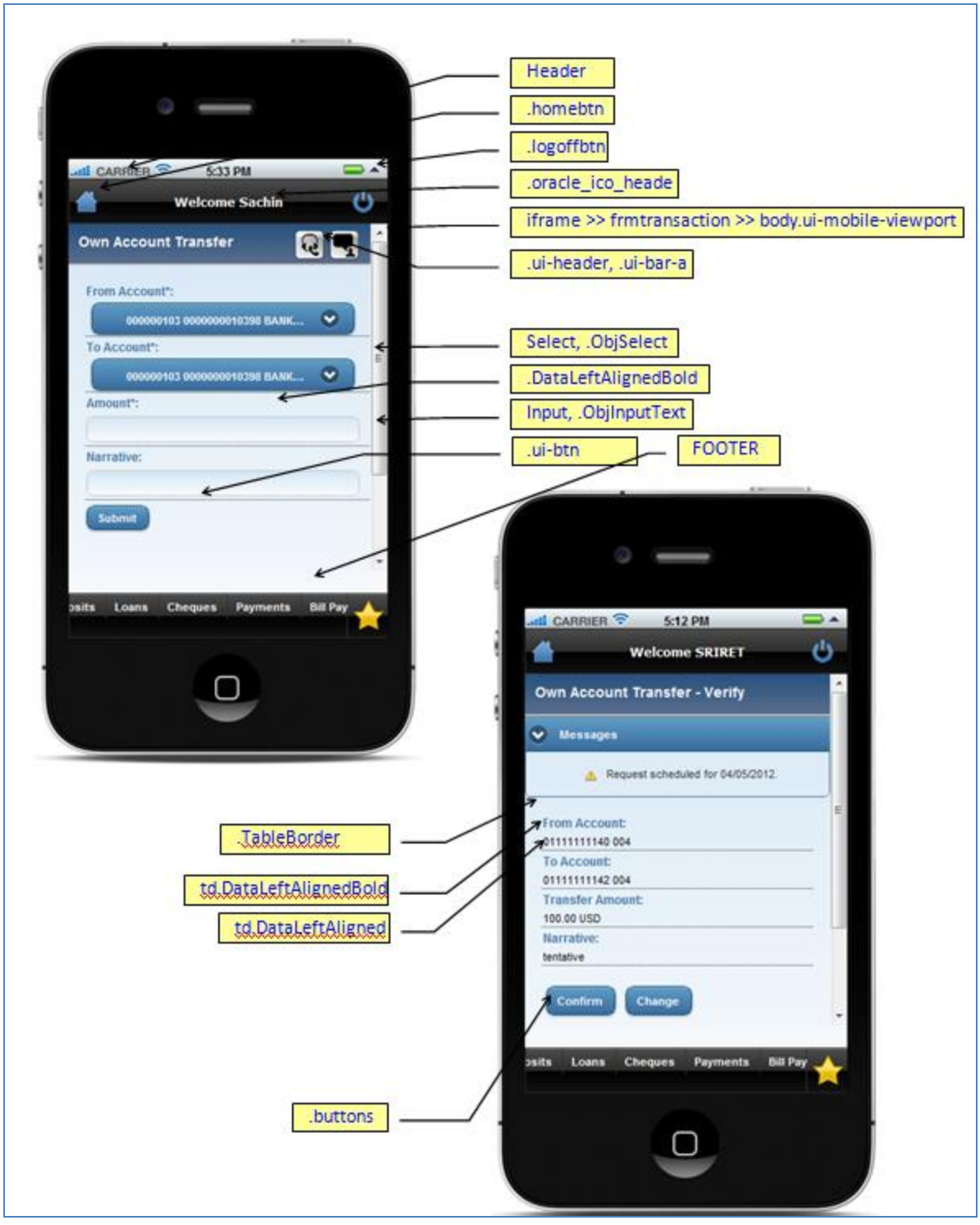
```
<link REL="stylesheet" TYPE="text/css" HREF="css/android/ THMDEFMOB/eng_42.css "></link>
```

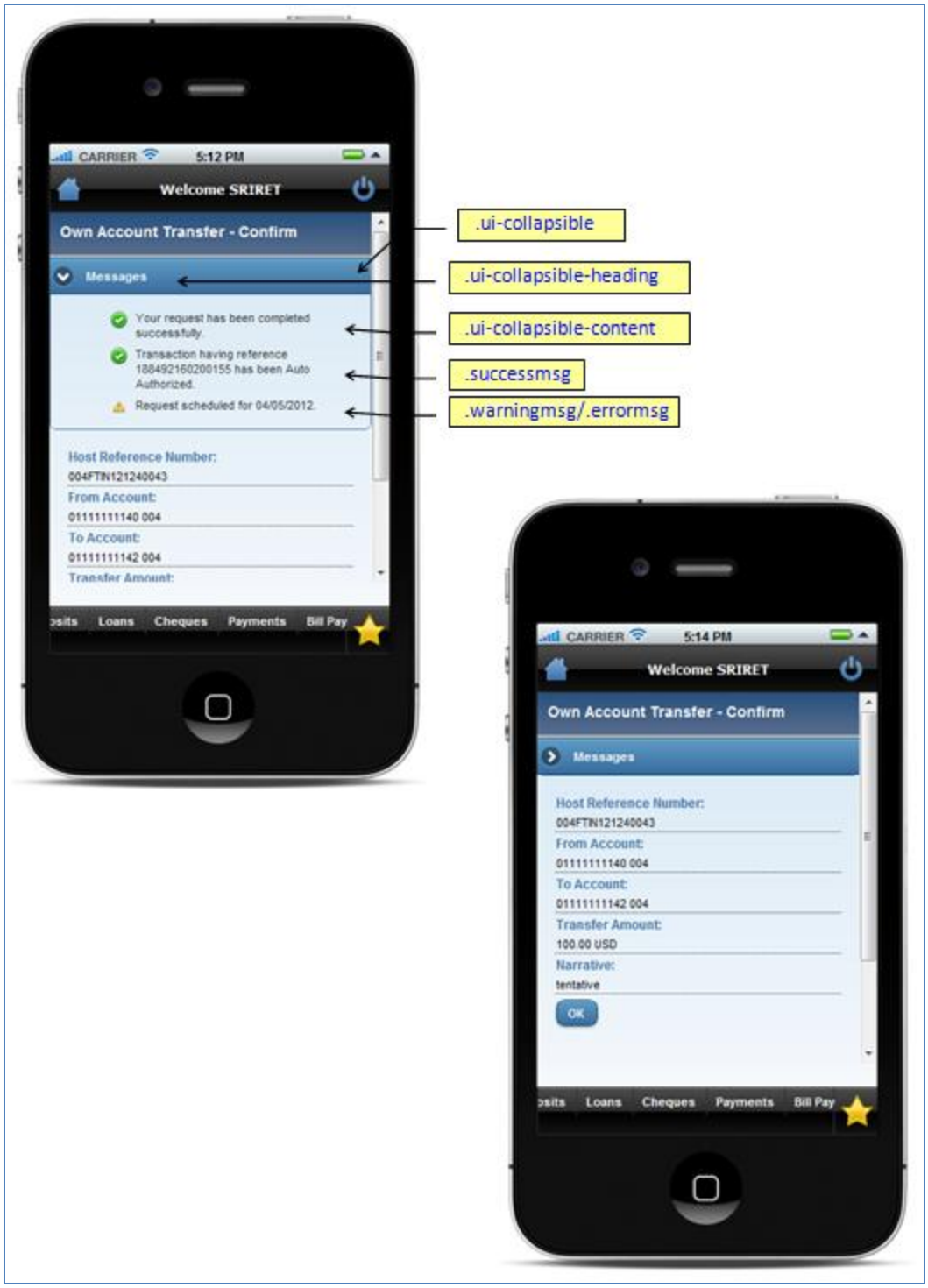
Orientation: Portrait

The classes shown in the figures below are defined in eng_42.css.

Login Screen

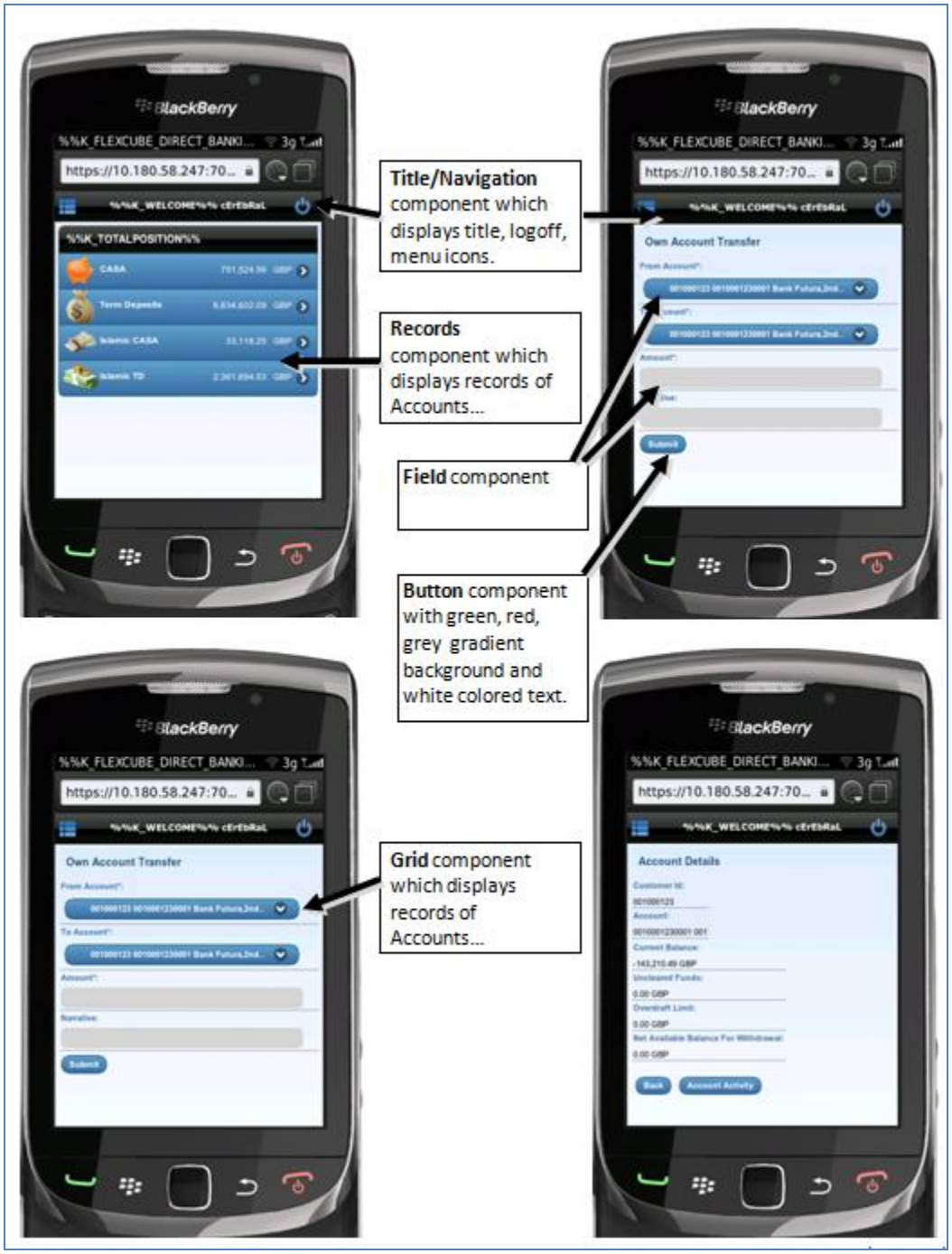
Transaction Screen – Forms, Verification (Read Only)


Header
.homebtn
.logoffbtn
.oracle_ico_heade
iframe >> frmtransaction >> body.ui-mobile-viewport
.ui-header, .ui-bar-a
Select, .ObjSelect
.DataLeftAlignedBold
Input, .ObjInputText
.ui-btn
FOOTER
CARRIER
5:33 PM
Welcome Sachin
Own Account Transfer
From Account*:
To Account*:
Amount*:
Narrative:
Submit
Loans
Cheques
Payments
Bill Pay
CARRIER
5:12 PM
Welcome SRIRET
Own Account Transfer - Verify
Messages
Request scheduled for 04/05/2012.
From Account:
01111111140 004
To Account:
01111111142 004
Transfer Amount:
100.00 USD
Narrative:
tentative
Confirm
Change
Loans
Cheques
Payments
Bill Pay
.TableBorder
td.DataLeftAlignedBold
td.DataLeftAligned
.buttons


CARRIER
5:12 PM
Welcome SRIRET
Own Account Transfer - Confirm
Messages
Your request has been completed successfully.
Transaction having reference 188492160200155 has been Auto Authorized.
Request scheduled for 04/05/2012.
Host Reference Number:
004FTN121240043
From Account:
01111111140 004
To Account:
01111111142 004
Transfer Amount:
Loans
Cheques
Payments
Bill Pay
.ui-collapsible
.ui-collapsible-heading
.ui-collapsible-content
.successmsg
.warningmsg/.errormsg
CARRIER
5:14 PM
Welcome SRIRET
Own Account Transfer - Confirm
Messages
Host Reference Number:
004FTN121240043
From Account:
01111111140 004
To Account:
01111111142 004
Transfer Amount:
100.00 USD
Narrative:
tentative
OK
Loans
Cheques
Payments
Bill Pay

Blackberry Components

These scripts are typically hard-wired to the particular page; if the content of the page changes, the scripts may have to be rewritten, since there is no way of abstracting one from the other. HTML Components provide this ability. Following are some examples of components.



Blackberry Login Page

Below is a screen shot of the Login page. It consists of the logo which is placed on the top center, a user ID and password text fields, a sign in button.

1) **Oracle Logo:** The Logo is placed in a div at the top left corner with class name for logo is “oracle_logo_login” and title text class name is “logotext”.

2) **Login Section:** The Login section is placed in a table with class name is ”MainLoginTable”. Text input fields for User ID and Password is placed on the left .

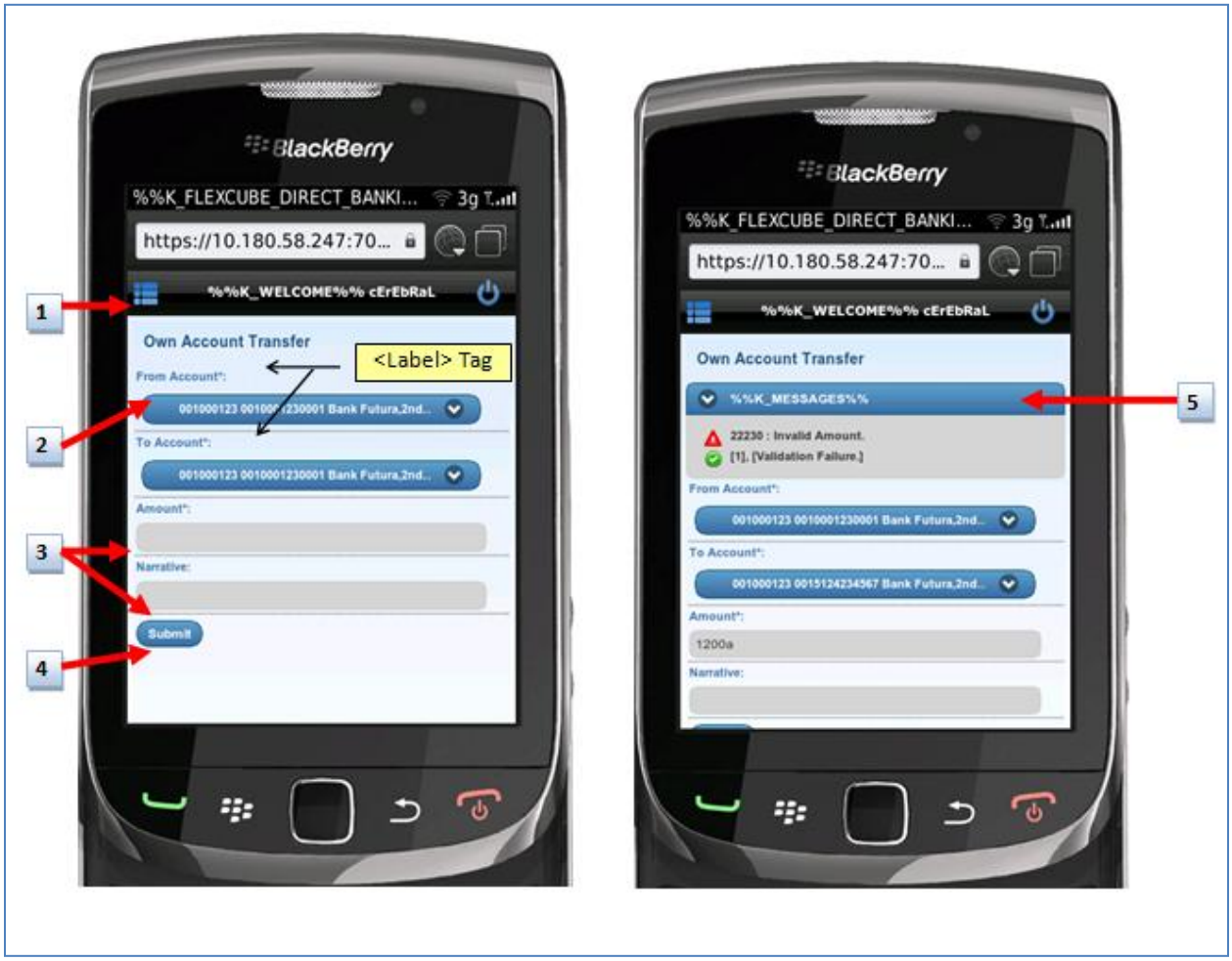
Blackberry Landing Screen

Once the user has logged in, the FLEXCUBE application menu appears.

1) **Menu:** Shows menu.
2) **Welcome Details :** Welcome user details placing on the top bar.
3) **Log off :** Logout from current screen.
4) **Total Positions:** Shows list of various account types with balance details .
5) **Account Type Details :** Shows list of account details (shows CASA details in Fig.3.3.1c).
6) **Back :** Return back to Total Position screen

Blackberry Components – IFrame, Input, Combobox, Error Message

After logging in, transactionpanel div tag displays iframe (1) in which main transaction screen appears with Combobox (2), Input Text Field (3), buttons (4), and error message dialog (Collapse/Expand) (5).



Cascading Style Sheets

Style Sheets of the FLEXCUBE application are available in the “css” folder. Only one styles sheets are maintained for both the Login page and the Transaction pages.

- Login Page and Transaction Pages : eng_42.css

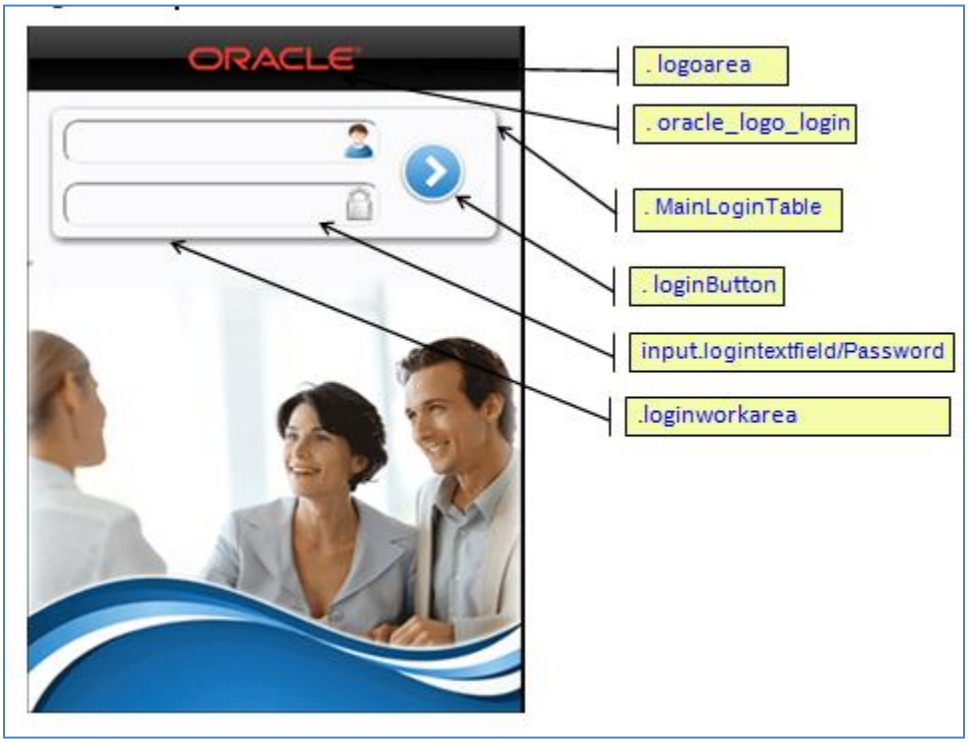
In the Login and Transaction page, the CSS is called in the head section as given below, when user agent set to iphone, blackberry, android or nokia, path for CSS also change resp.

```
<head>
   <link REL="stylesheet" TYPE="text/css" HREF="css/blackberry/THMDEFMOB/eng_42.css "></link>
</head>
```

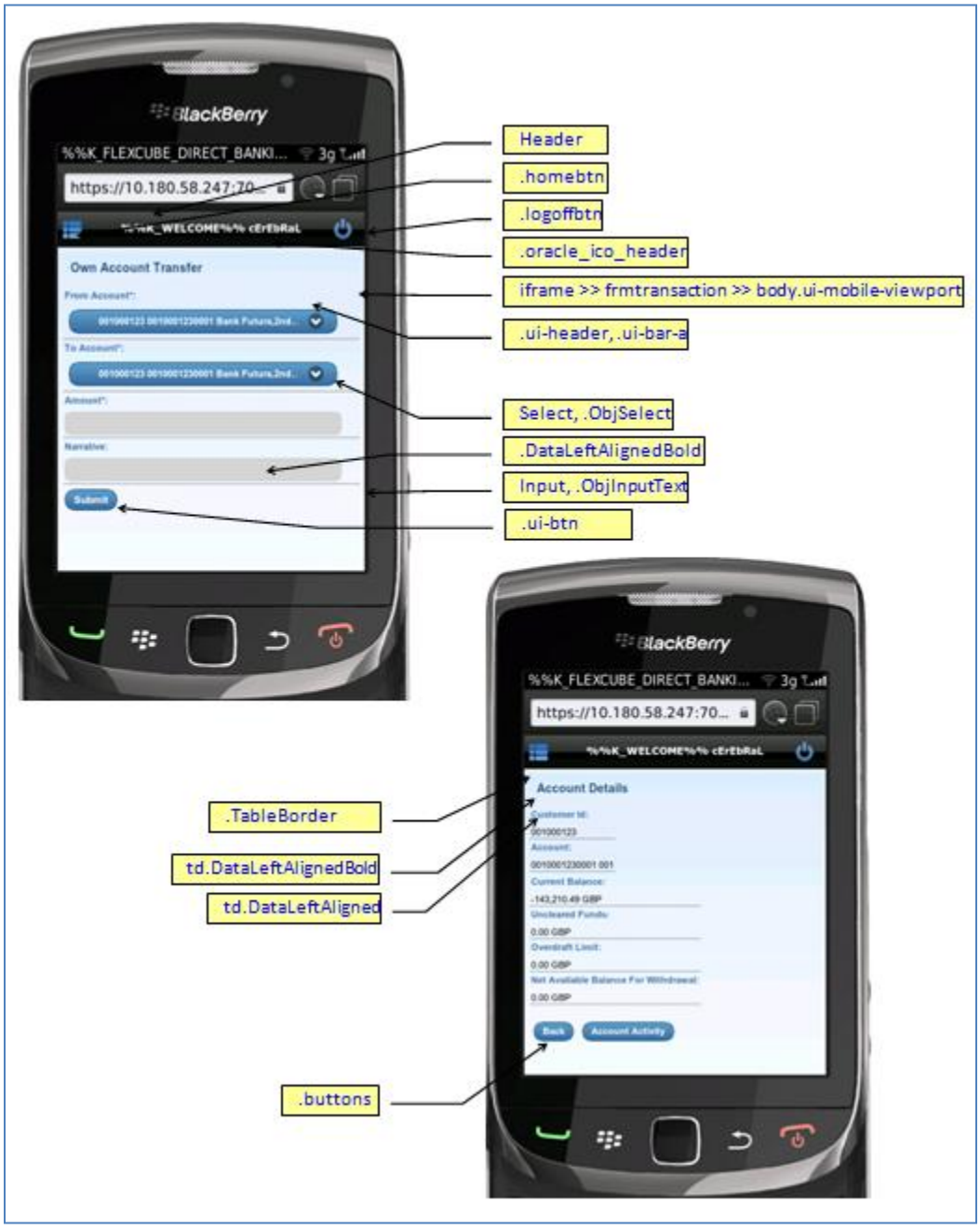
Orientation : Portrait

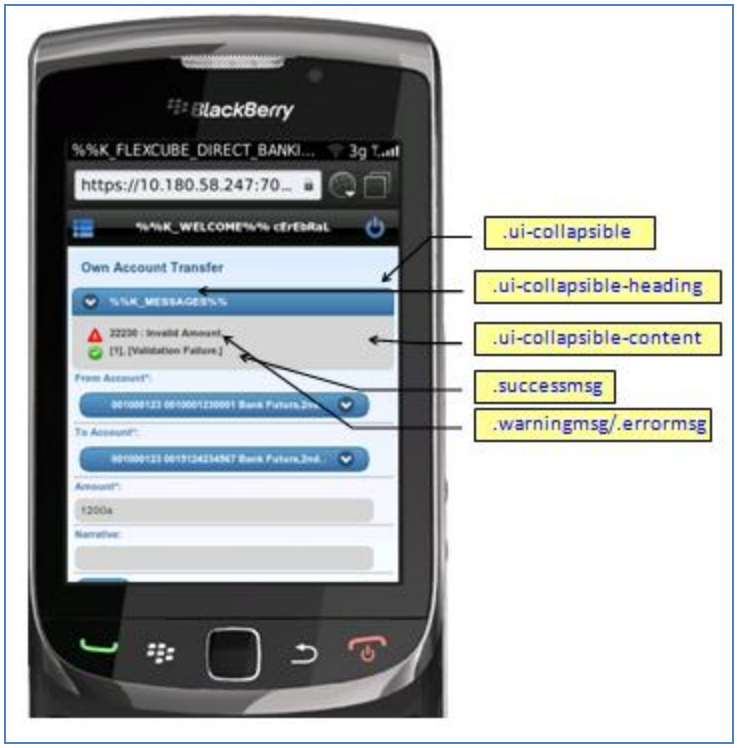
The classes shown in the figures below are defined in eng_42.css.

Login Screen



Transaction Screen – Forms, Verification (Read Only)





Javascript Files

JavaScript is used across the application to create functionalities like, menu dropdowns, calendar popup, form validations, and much more.

JavaScript files are stored in the “jsdir” and jqtouch folder. These files like CSS files are called within the Head tag as given below.

```
<head>
<script src="jsdir/common.js" type="text/JavaScript" language="JavaScript"  charset="utf-8"></script>
<script src="jqtouch/iscroll.js?v3.7.1" type="text/javascript" charset="utf-8"></script>
<script src="jqtouch/jqscroll.js"  type="text/javascript" charset="utf-8"></script>
<script src="jqtouch/jqtouch.min.js" type="text/javascript"  charset="utf-8"></script>
<script type="text/javascript" src="JS/jquery-1.6.4.min.js"></script>
<script type="text/javascript" src="JS/jquery.mobile-1.0.1.min.js"></script>
</head>
```

We have predefined JavaScript files like scrolling horizontally and vertically, touch effect on screen, etc. which behave like modules and can be reused when ever required. In order to use these JS files, predefined IDs must be provided for the HTML tags. For example, when we are creating a virtual keyboard, we first need to import the JavaScript file as given below

```
<head>
<script type="text/JavaScript" language="JavaScript" src=" jqtouch/jqscroll.js "></script>
</head>
```

And then Provide the IDs in the HTML tags as below

```
<div id="wrapper_l" class="wrapper_l">
        <div id="scroller_l" class="scroller_l">
                <table cellpadding="0" cellspacing="0">
                        <tr>
                        <td>
                                ...
                                ...
                        </td>
                        </tr>
                </table>
        </div>
</div>
```

This will take care of the functionality of a scrolling vertically and horizontally on touch.

Images stored in folder named Mobile in which there are some common images.

Oracle_small.png, logout.png, menuBlue.png, changePass.png – These images are transparent and can be place on any kind of background.

Browser Support

Oracle FLEXCUBE Direct Banking is compliant with the popular browsers in the industry for internet banking. The user interface for Internet Banking follows standard XHTML 1.0 STRICT guidelines and any web browser and user agent supporting the standard can be used for accessing Oracle FLEXCUBE Direct Banking user interface.

The following browsers are typically used for testing purposes

Microsoft Internet Explorer 6.0+

Microsoft Internet Explorer 7.0+

Mozilla / Firefox 1.0+, 2.0+, 3.0+

Applet Safari 3.0+

Google Chrome 1.0+

The above browsers can be supported on Windows, Linux and Macintosh Operating Systems.

Browser Based Mobile Banking on iPad - User Interface Design

The browser based mobile banking for iPad is same as the Internet banking channel. The mobile.jsp as described in Section 3.3.1, checks if the 'User Agent' is iPad and then redirects to ENULogin.jsp

In order to optimize the screens for iPad Safari browser, following changes have been done:

Viewport Metadata

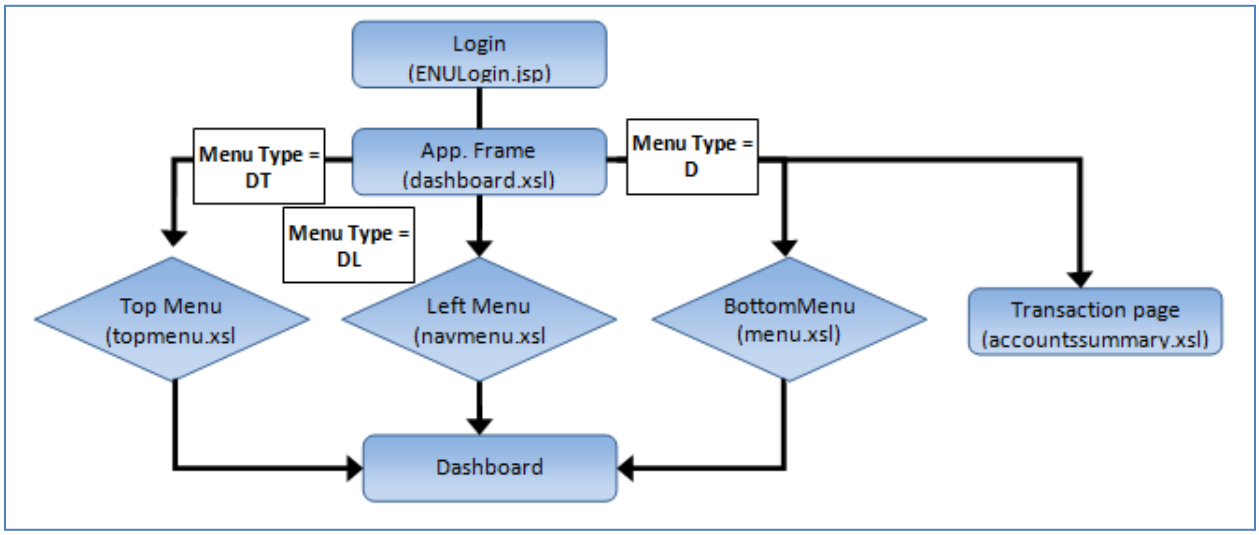
iPad web apps support the viewport metadata (placed in ENULogin.jsp and RFrame.xsl) to give information about the zoom and content area that you want to use in your website.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"></meta>
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0,
minimum-scale=1.0, maximum-scale=1.0" ></meta>
<meta name="apple-mobile-web-app-capable" content="yes" ></meta>
<meta name="apple-mobile-web-app-status-bar-style" content="default" ></meta>
<meta name="HandheldFriendly" content="true"></meta>
<meta name="MobileOptimized" content="width"></meta>
```

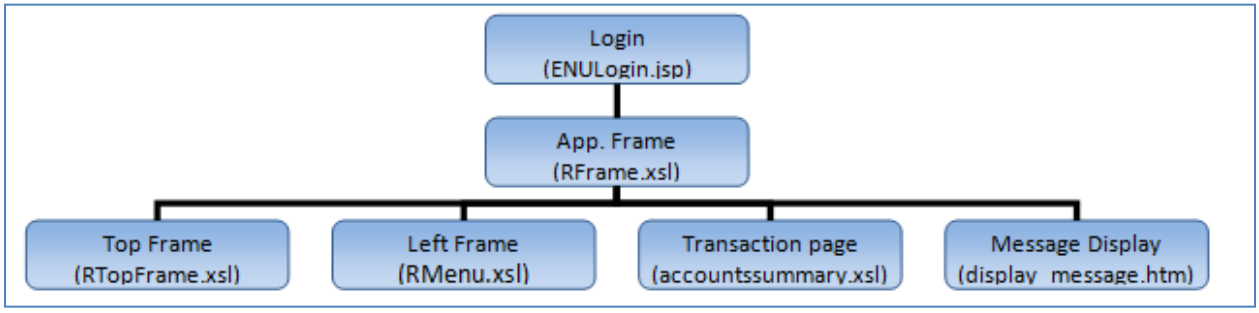
Static HTML Content

In this topic, we will discuss the HTML content, such as menus, forms, themes, tables, grids, graphs etc. that have been used across the FLEXCUBE application. For a better understanding, below are structures of the application

Contemporary Version:

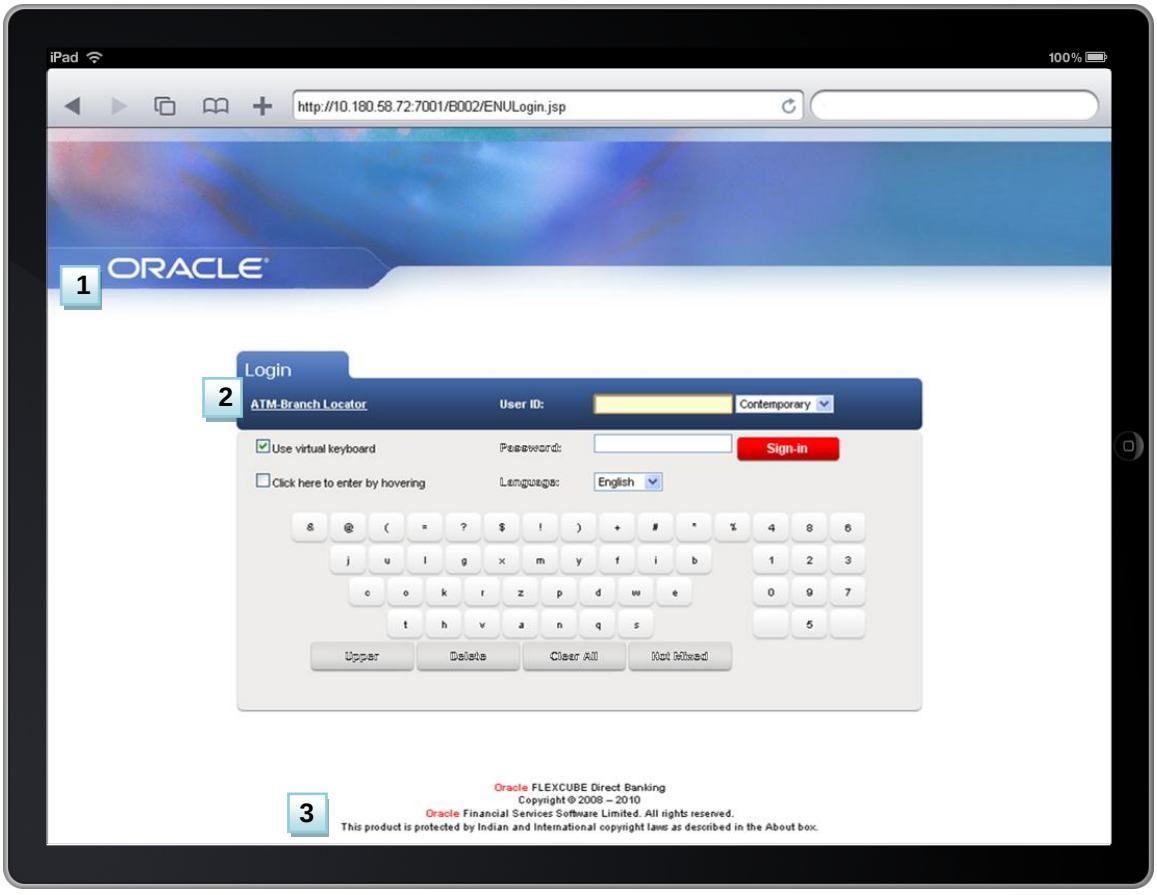


Classic Version:



Login Page

Below is a screen shot of the Login page. It consists of the logo which is placed on the top center, a user ID and password text fields, a sign in button.



1) **Oracle Flexcube Logo:** The Logo is placed in a div at the top left corner with class name for logo is “oracle_logo_login” and title text class name is“logotext”.

2) **Login Section:** The Login section is placed in a table with class name is ”MainLoginTable”. Text input fields for User ID and Password is placed on the left .

3) **Footer:** Footer section is placed at the bottom which has a company product’s copyright details.

Cascading Style Sheets

Style Sheets of the FLEXCUBE application are available in the “css” folder. Only one styles sheets are maintained for both the Login page and the Transaction pages.

- Login Page and Transaction Pages : eng_42.css

In the Login and Transaction page, the CSS is called in the head section as given below, when user agent set to ipad path for CSS also change resp.

```
<link REL="stylesheet" TYPE="text/css" HREF="css/ipad/eng_42.css "></link>
```

We have predefined CSS to control scrolling vertically with two finger on dashboard and smooth scroll in transaction window panel and UI download table etc. which behave like modules and can be reused when ever required. In order to use these CSS files, predefined IDs and class names must be provided for the HTML tags.

theme.css

```
.mainFrame

 { display: block; position: fixed; background-position:fixed; background-attachment:scroll; overflow-y:scroll}

.mainscroll { display: block; height: 90%;  -webkit-perspective: 1000; -webkit-backface-visibility: hidden;}

.buttons {              font-size:12px; font-weight:bold; padding:5px !important;  height:26px !important; }


/* Portrait */

@media screen and (orientation:portrait)

{

 .contentarea { height:100%; }

 .Buttons {font-size:12px;  font-weight:bold; padding:3px 5px 10px 5px !important; height:30px!important;  }

 .buttons {font-size:12px; font-weight:bold; padding:5px !important; height:26px !important; }

 .column-large { margin:0 1.1em 0 1.1em !important;}

 .displaytable {overflow: auto; overflow-y: scroll; overflow-x: scroll; overflow:-moz-scrollbars-horizontal;}

}

/* Landscape */
```

```
@media screen and (orientation:landscape)

{

  .contentarea { height:100%; }

  .Buttons {font-size:12px; font-weight:bold; padding:3px 5px 10px 5px!important; height:30px!important;    }

  .buttons { font-size:12px; font-weight:bold; padding:5px !important; height:26px !important;       }

  .column-large { margin:0 1.1em 0 1.1em !important;}

  .displaytable {overflow: auto; overflow-y: scroll; overflow-x: scroll; overflow:-moz-scrollbars-horizontal;}

}
```

Topmenu.css

```
.contentMenu {-webkit-user-select: none;  /* prevent copy paste for all elements */}
```

Javascript Files

JavaScript is used across the application to create functionalities like, menu dropdowns, calendar popup, form validations, and much more.

JavaScript files are stored in the “jsdir” and jqtouch folder. These files like CSS files are called within the Head tag as given below.

```
<head>
<script src="jqtouch/jquery.1.3.2.min.js" type="application/x-javascript" charset="utf-8"></script>
<script src="jsdir/common.js" type="text/JavaScript" language="JavaScript"  charset="utf-8"></script>
<script src="jqtouch/iscroll.js?v3.7.1" type="text/javascript" charset="utf-8"></script>
<script src="jqtouch/jqscroll.js"  type="text/javascript" charset="utf-8"></script>
<script src="jqtouch/jqtouch.min.js" type="text/javascript"  charset="utf-8"></script>
</head>
```

We have predefined JavaScript files like scrolling horizontally and vertically, touch effect on screen, etc. which behave like modules and can be reused when ever required. In order to use these JS files, predefined IDs must be provided for the HTML tags. For example, when we are creating a virtual keyboard, we first need to import the JavaScript file as given below

```
<head>
<script type="text/JavaScript" language="JavaScript" src=" jqtouch/jqscroll.js "></script>
</head>
```

And then Provide the IDs in the HTML tags as below

```
<div id="wrapper_l" class="wrapper_l">
        <div id="scroller_l" class="scroller_l">
                <table cellpadding="0" cellspacing="0">
                        <tr>
                        <td>
                                ...
                                ...
                        </td>
                        </tr>
                </table>
        </div>
</div>
```

This will take care of the functionality of a scrolling vertically and horizontally on touch.

We have predefined JavaScript (placed in jqueryinclude.xsl) like scrolling vertically with two finger on dashboard and smooth scroll in transaction window panel and UI download table etc. which behave like modules and can be reused when ever required. In order to use these JS files, predefined IDs must be provided for the HTML tags. We first need to import the JavaScript file in to jqueryinclude.xsl as given below

```
<head>
/* used for scrolling horizontally and vertically in uidownload table */
<script type="text/JavaScript" src='JS/touchscroll.js'></script>

/* used for scrolling vertically in dashboard screen and transaction window */
<script type="text/JavaScript" src="JS/touch-scroll.min.js"></script>
</head>
```

Javascript function code in jqueryinclude.xsl >>>

```
$(document).ready(function() {


if(useragent=="ipad"){

    if (!parent.isMenuClassic())

    {

        if (parent.frames.length > 0)
```

```
        {
            var heightDiv = $('form[name="frmmain"]').height() + $(window).height() - 200;
            $('.contentarea').css({height:heightDiv});
            $('body').addClass("mainFrame");
            $('form[name="frmmain"]').addClass("mainscroll");
            try
            {
                $('form[name="frmmain"]').touchScroll({elastic: true, momentum: false});
                touchTableScroll('displayTable');
                function loaded()
                {
                document.addEventListener('touchmove', function(e){ e.preventDefault(); });
                myScroll = new iScroll('displayTable', {elastic: false, momentum: false});
                }

            document.addEventListener('DOMContentLoaded', loaded);

            }
            catch(e){
            }
        }
    }

}});
```

J2ME Client - User Interface Design

Rich version of j2me client (J2ME RICH) supports pluggable themes similar to CSS and somewhat simpler than Swing's pluggable Look And Feel. This document covers how to create a theme file (.res)

Every LWUIT component has a style associated with it this style can be manipulated manually and can be customized using a set of definitions for a specific component type. By defining some properties inside theme file (.**res**)

A theme file is very similar in spirit to CSS, yet it is much simpler and it is structured like a Java properties file. A theme file is comprised of key value pairs. The key acts in a similar way to a CSS selector that indicates the component or attribute affected by the theme value.

For example:

■ Button.font – font for all buttons

■ font – default application font applied to all components where no default is defined

The key element is comprised of an optional unique identifier ID for the component (the UIID) and a required attribute type. Unlike CSS, themes do not support elements such as hierarchy or more complex selectors.

Component UIIDs correspond to the component class name by convention.

For example.: Button, Label, CheckBox, RadioButton, Form, etcetera.

Theming Attributes

LWUIT components support many UI attributes

The supported attributes and their value syntax are illustrated below

bgAlign	Allows determining the alignment of a background image, only effective for non-scaled background images. Valid values include: BACKGROUND_IMAGE_ALIGN_TOP, BACKGROUND_IMAGE_ALIGN_BOTTOM, BACKGROUND_IMAGE_ALIGN_LEFT, BACKGROUND_IMAGE_ALIGN_RIGHT,

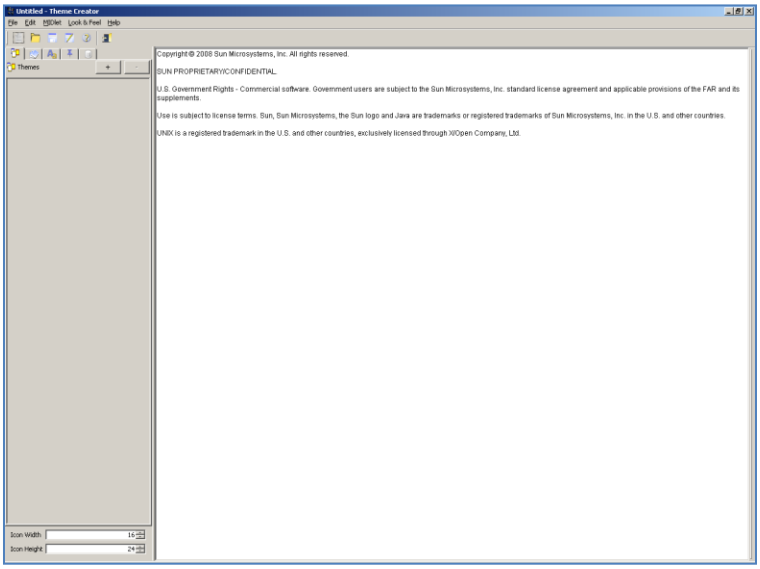
	BACKGROUND_IMAGE_ALIGN_CENTER
bgGradient	Determines the values for the gradient of the image. Accepts source/destination color as well as X/Y of the center of a radial gradient.
bgColor	Hexadecimal number representing the background color for the component in an unselected widget. For example, blue would be: ff
bgImage	Name of an image from within the resource that should be used as the background for this component. The image referenced must exist within the resource using the same name mentioned here
bgType	Allows determining the type of the background whether it is an image, color, or gradient. Valid values are: BACKGROUND_IMAGE_SCALED, BACKGROUND_IMAGE_TILE_BOTH, BACKGROUND_IMAGE_TILE_VERTICAL, BACKGROUND_IMAGE_TILE_HORIZONTAL, BACKGROUND_IMAGE_ALIGNED, BACKGROUND_GRADIENT_LINEAR_HORIZONTAL, BACKGROUND_GRADIENT_LINEAR_VERTICAL, BACKGROUUND_GRADIENT_RADIAL
fgColor	Hexadecimal number representing the foreground color for the component usually used to draw the font in an unselected widget. For example, red would be: ff0000
font	The name of the bitmap or system font from the build XML file
margin	The amount of margin for the component defined as 4 comma-separated integer values representing top, bottom, left, and right. For example, 1, 2, 3, 4 results in 1 pixel margin top, 2 pixels margin bottom, 3 pixels margin left and 4 pixels margin right.
padding	Padding is identical to margin in terms of format but it updates the padding property of the component.
transparency	A number between 0 and 255 representing the opacity of a component’s background. 0 means the background of the component doesn’t draw at all (fully transparent) while 255 represents a completely opaque background.

Creating Themes

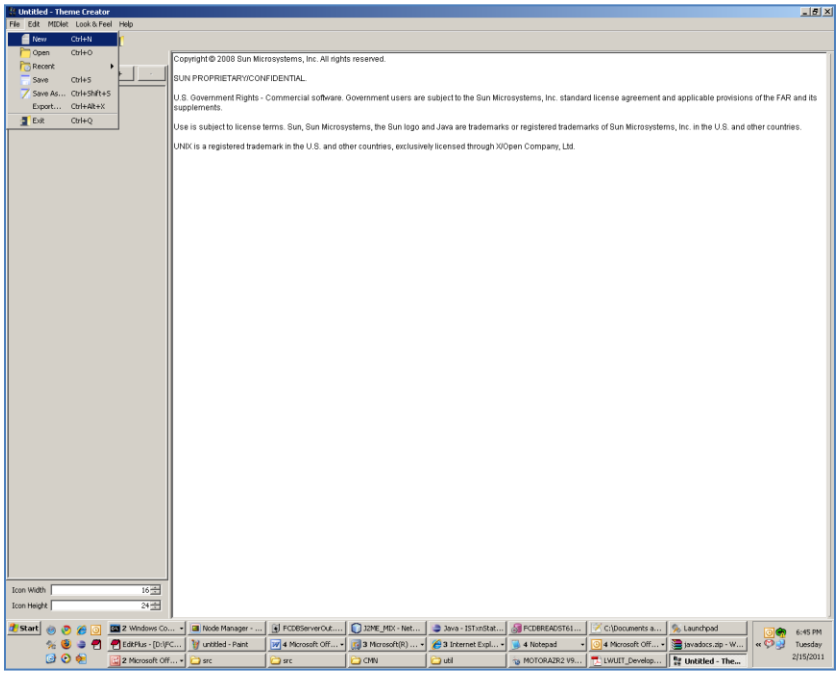
User can create theme files using a tool named LWUIT Theme Creator

Adding Simple Components

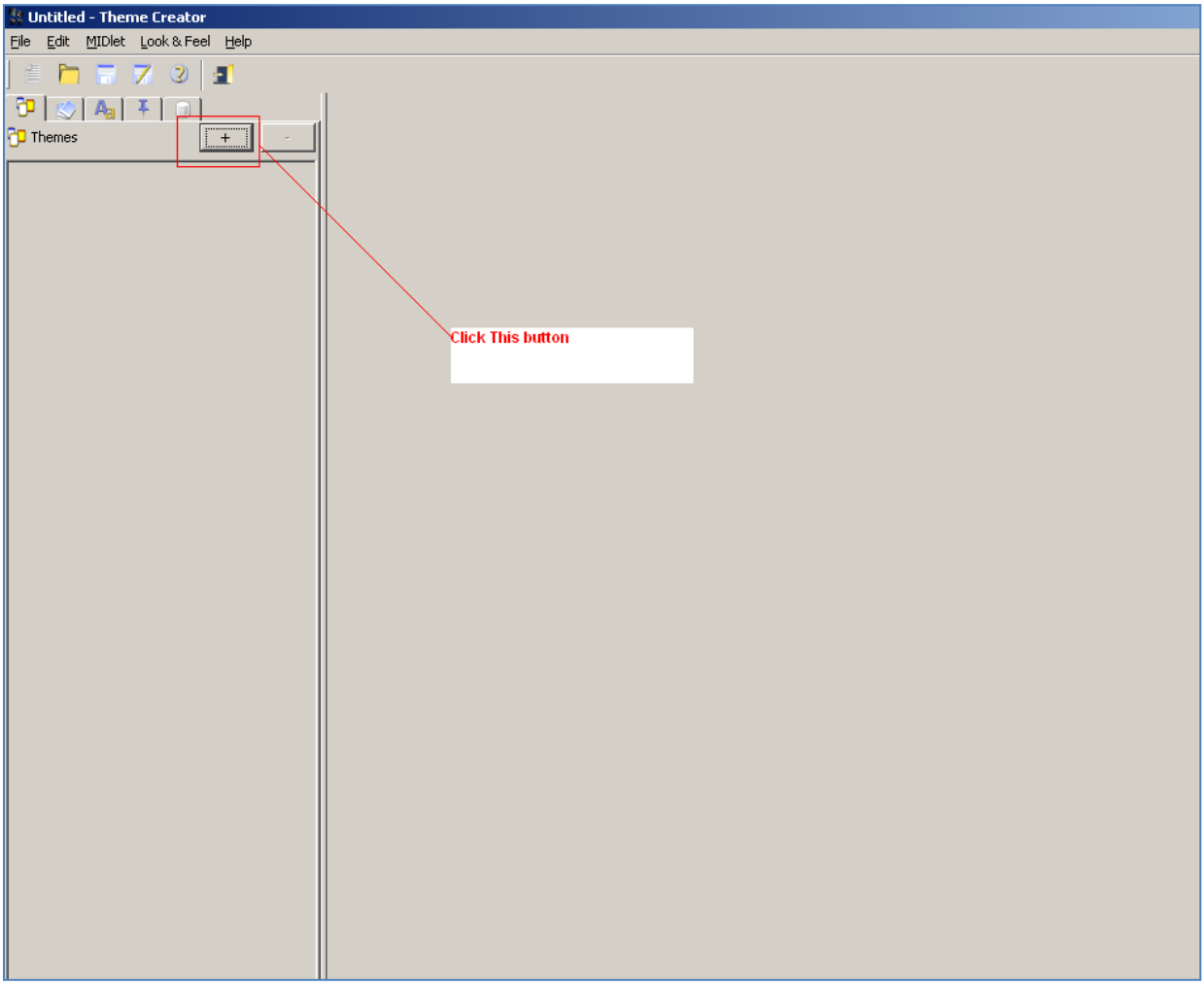
Open ResourceEdit.exe



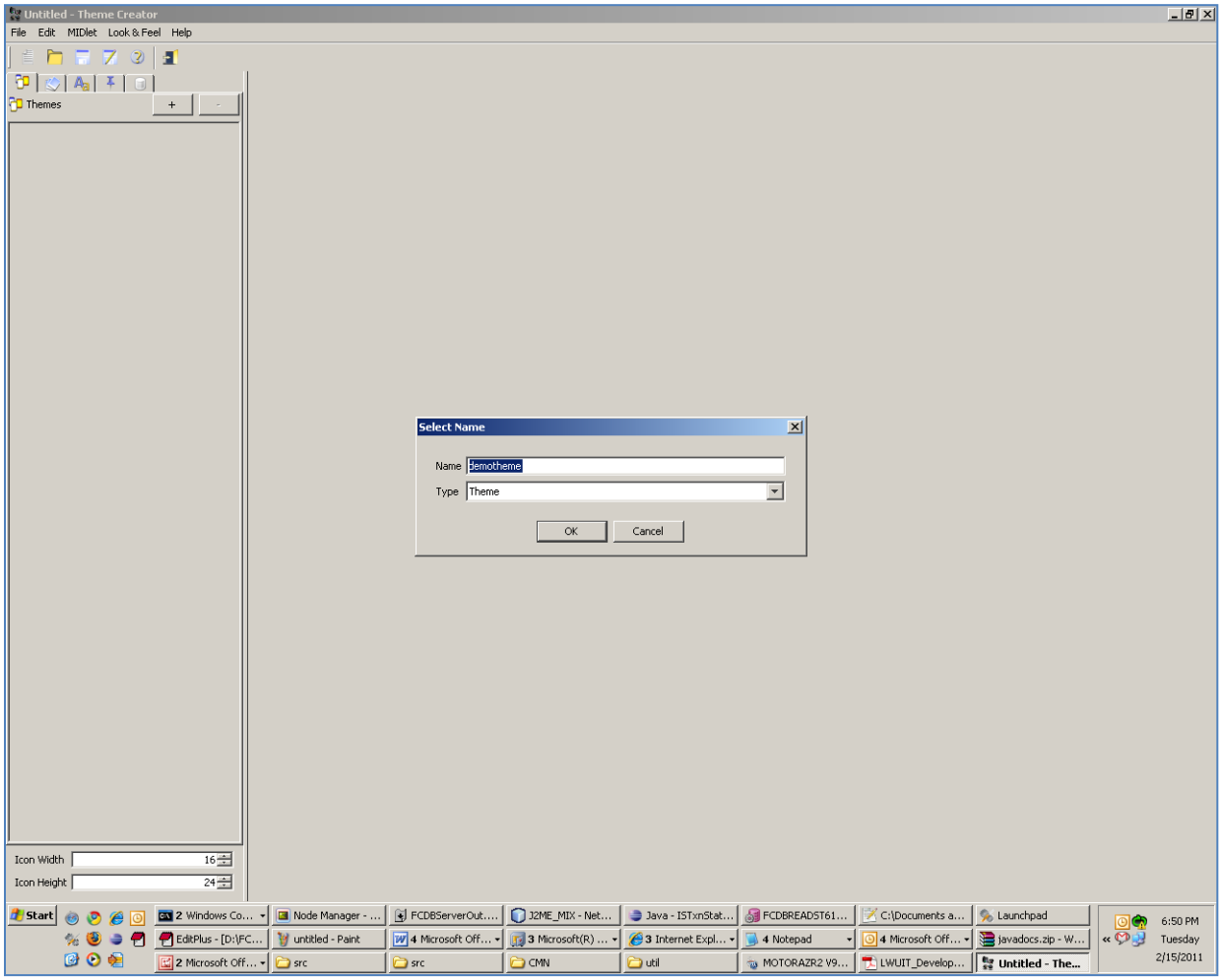
1. click File >New



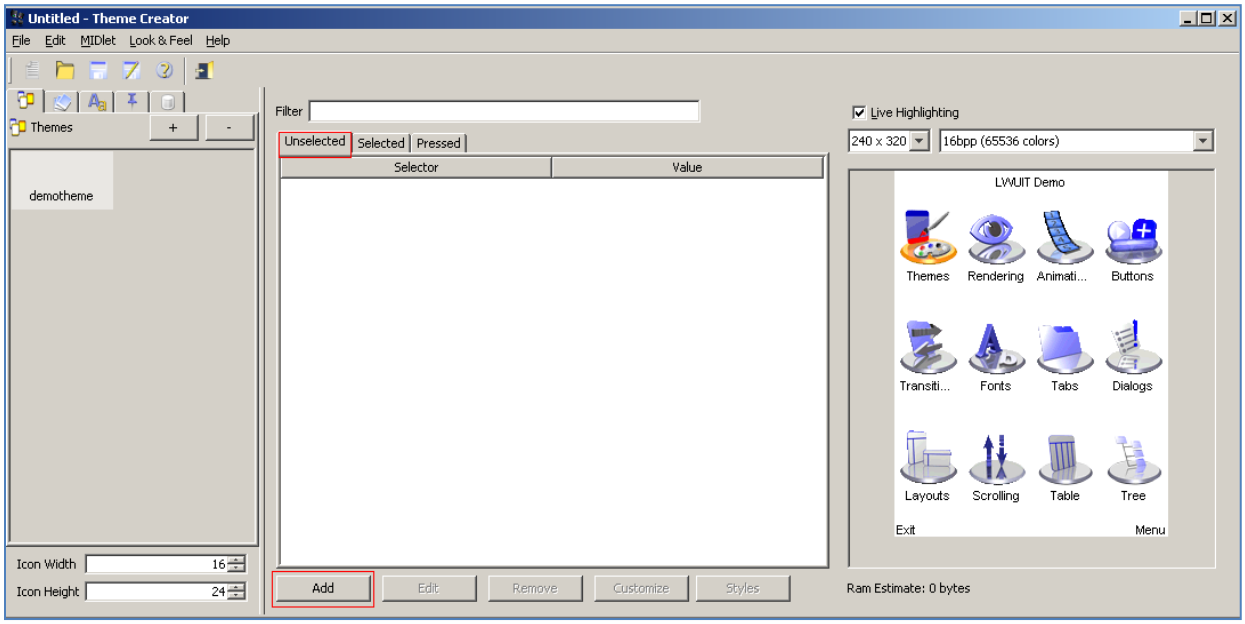
2. click + button as shown in below screen shot



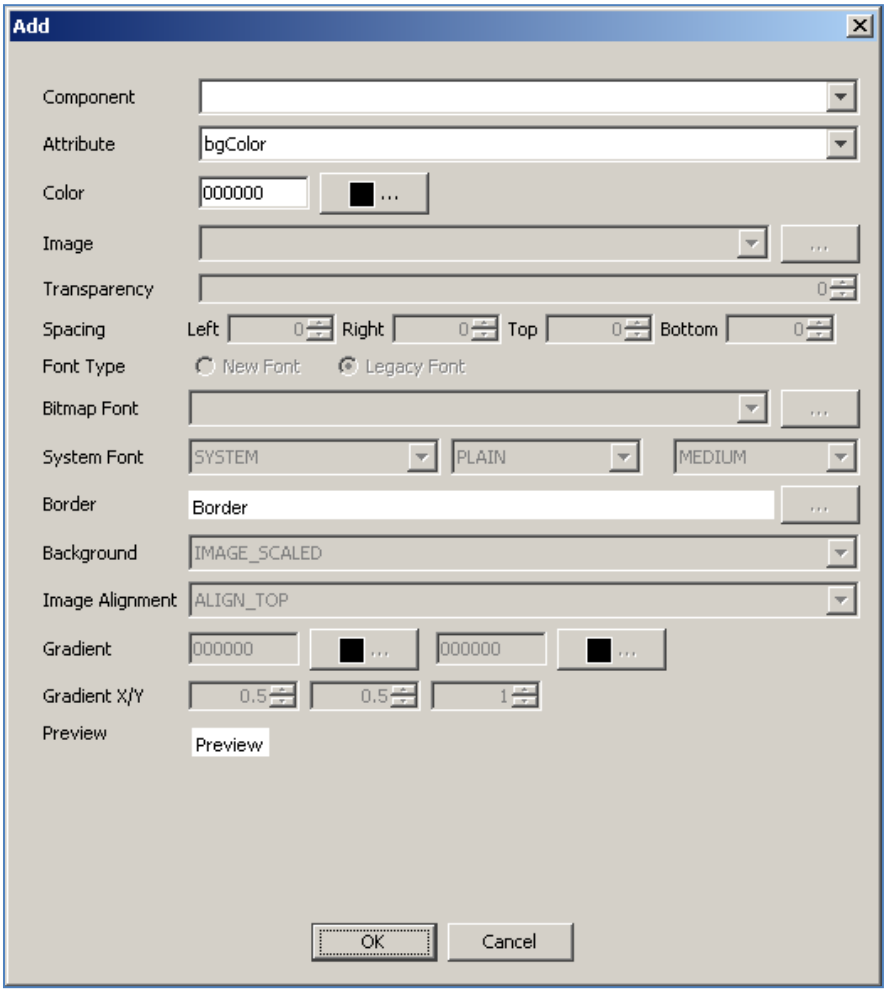
3. Now type the name of theme and click ok



4. Below screen will appear

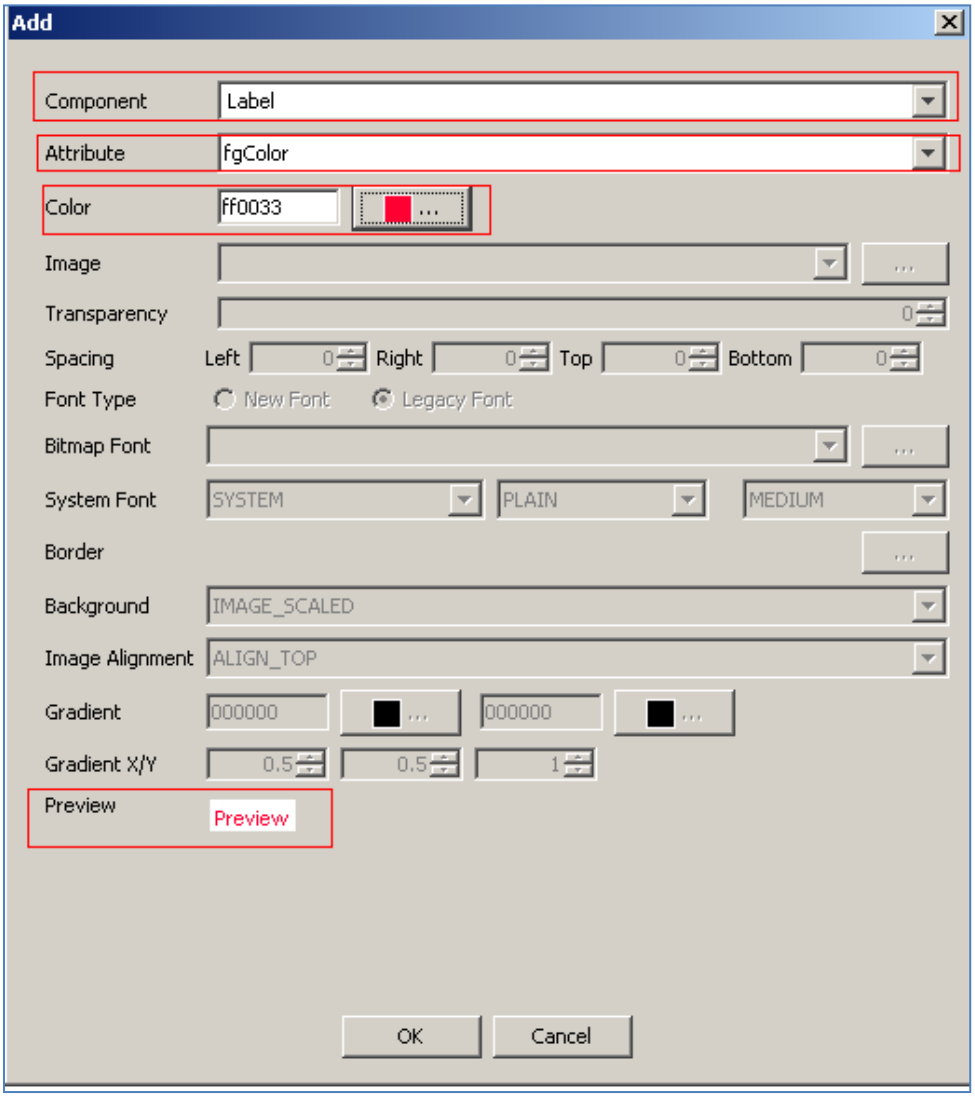


5. Now click on add button a new window will appear

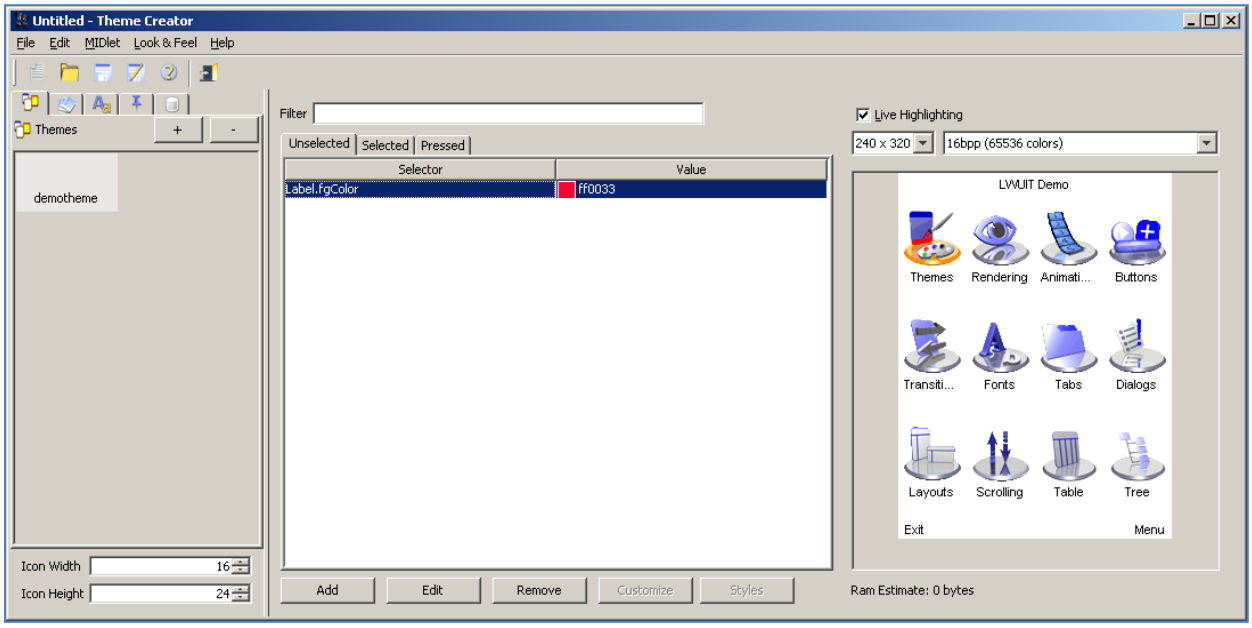


6. Now select a Component on which you want to apply theme (e.g. Label), Attribute (e.g. fgcolor)

And color and click ok



7. Now this property will be visible inside the editor

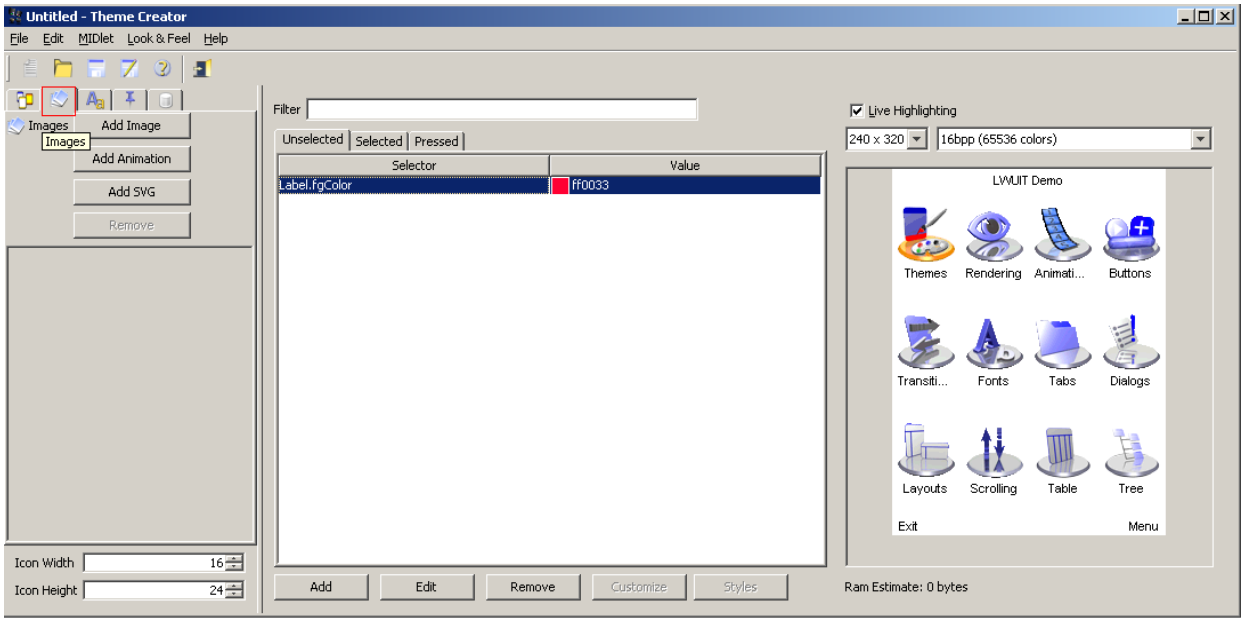


You can add many components' property by using this tool.

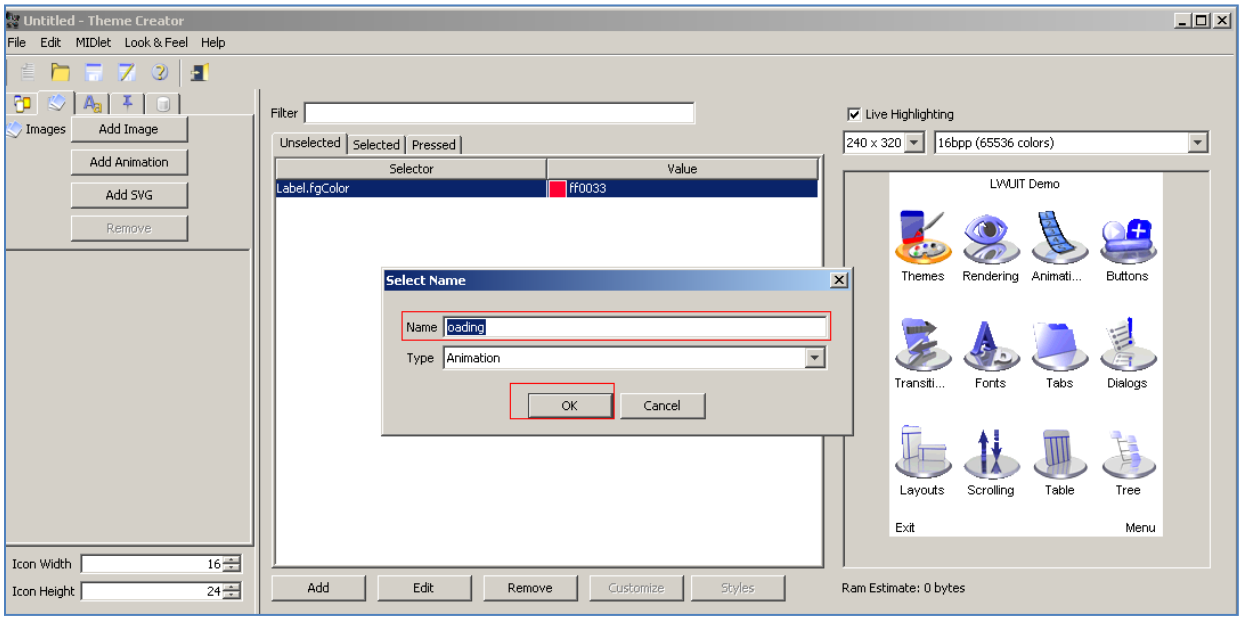
Adding GIF images

We can also add animation (GIF image files) using resource editor

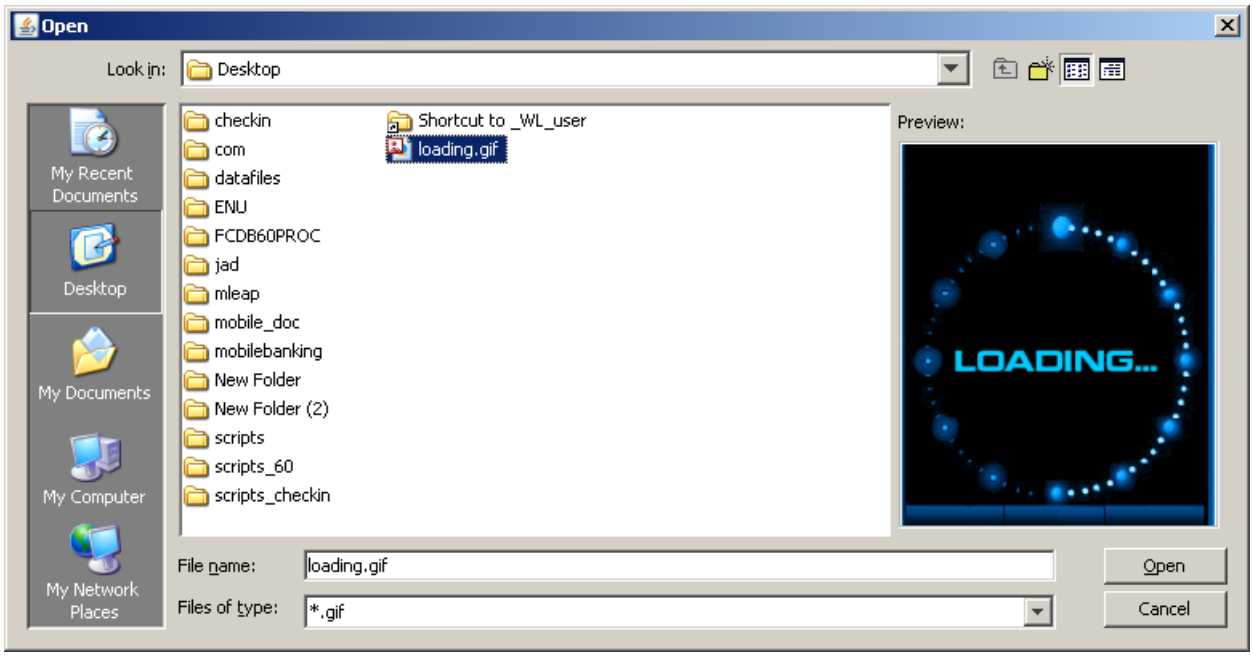
Click on the Image tab in theme creator as shown in following screen shot



Now click Add Animation button, type a name and click OK

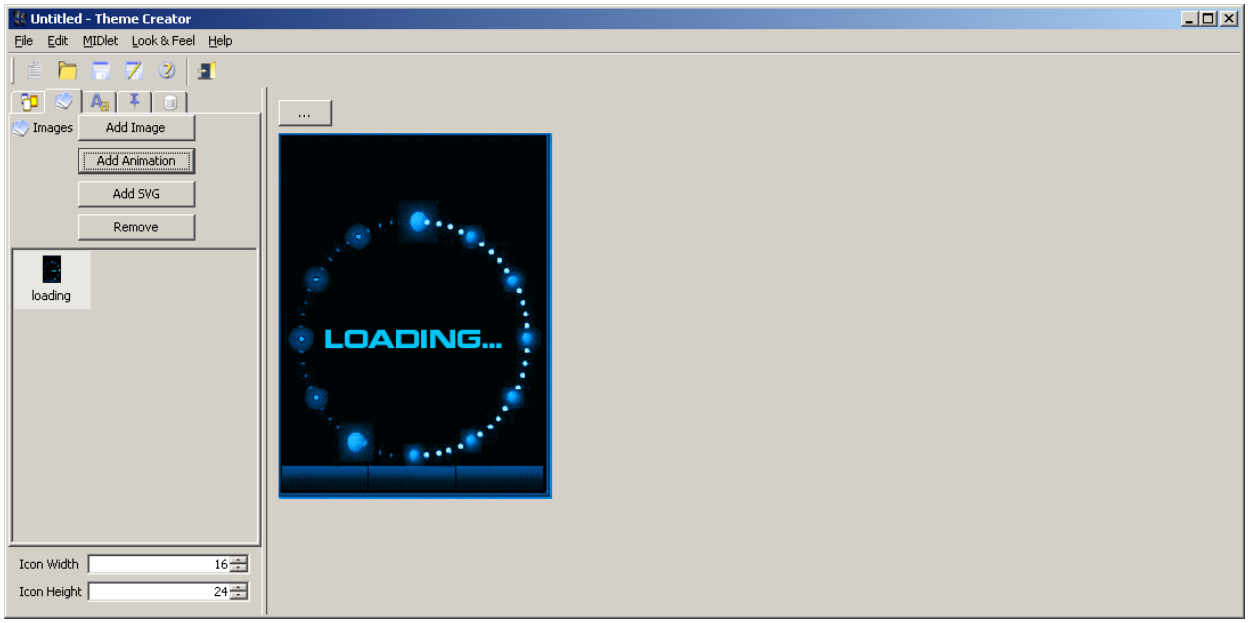


After clicking OK a new window will open



Select the .gif file from this window and click open, an animation will be added inside the theme file.

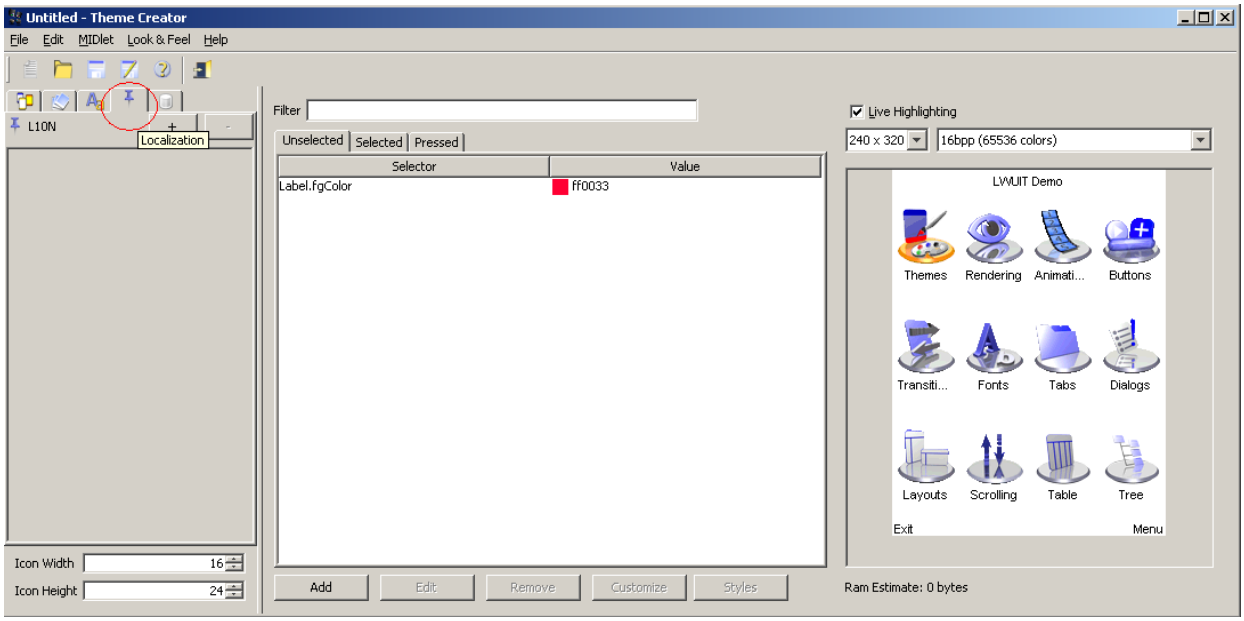
We can use this animation inside our application.



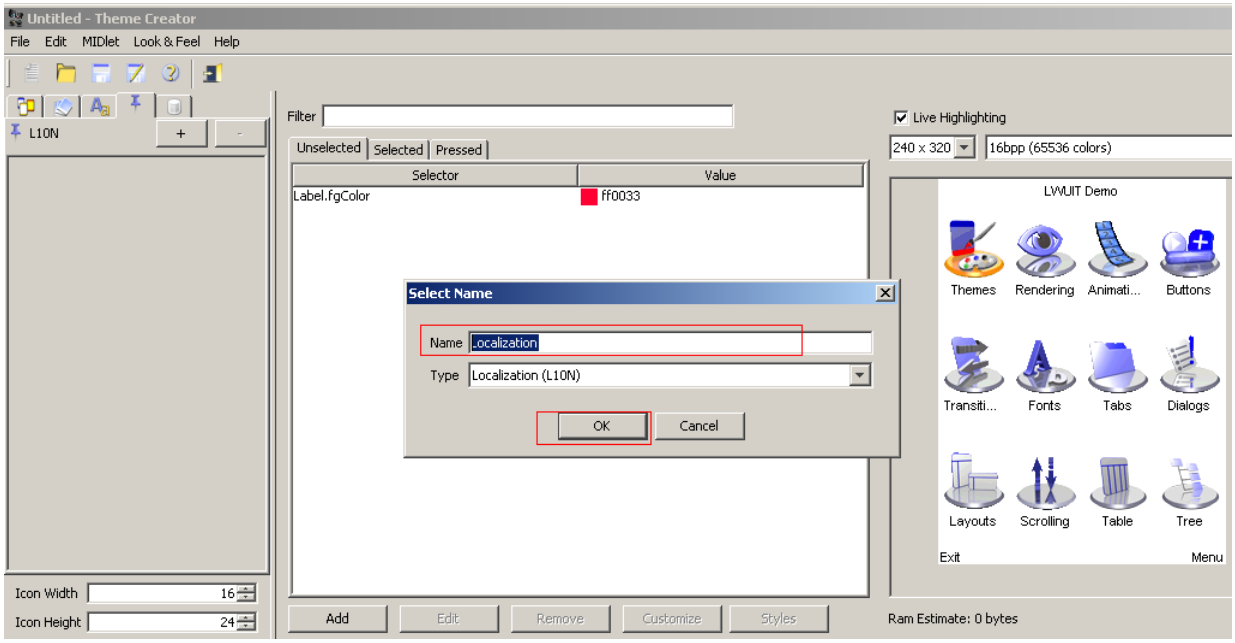
Adding Localization

A localization resource can be added by assigning key/value pairs to use within the application. A key can be mapped to a resource name in any locale

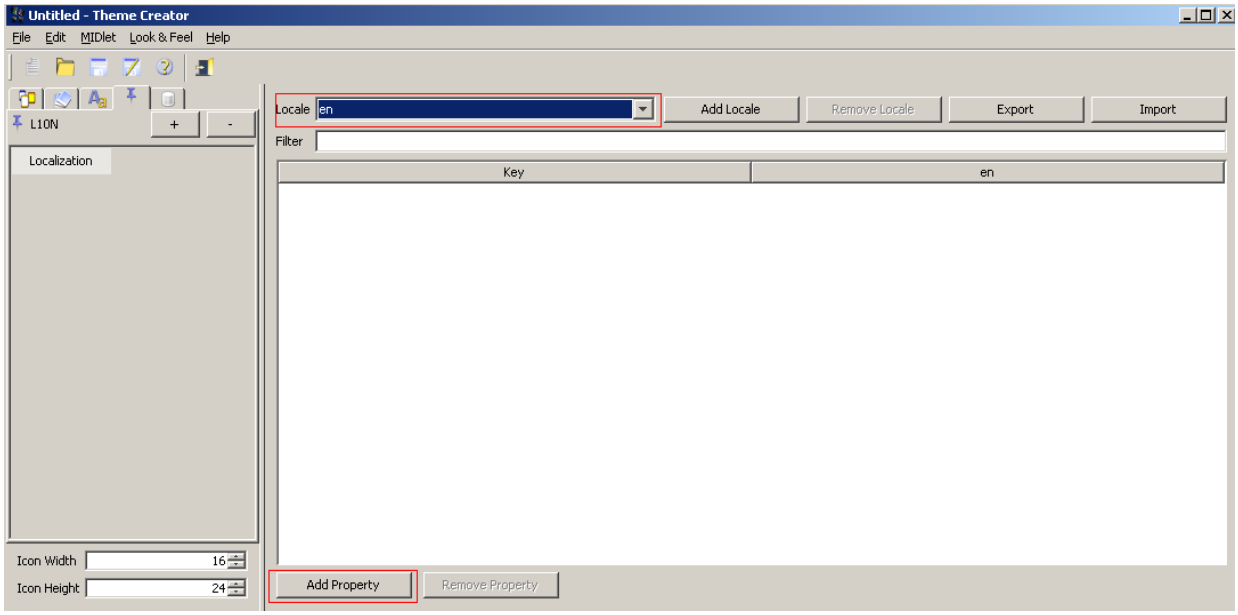
Select localization tab and click +



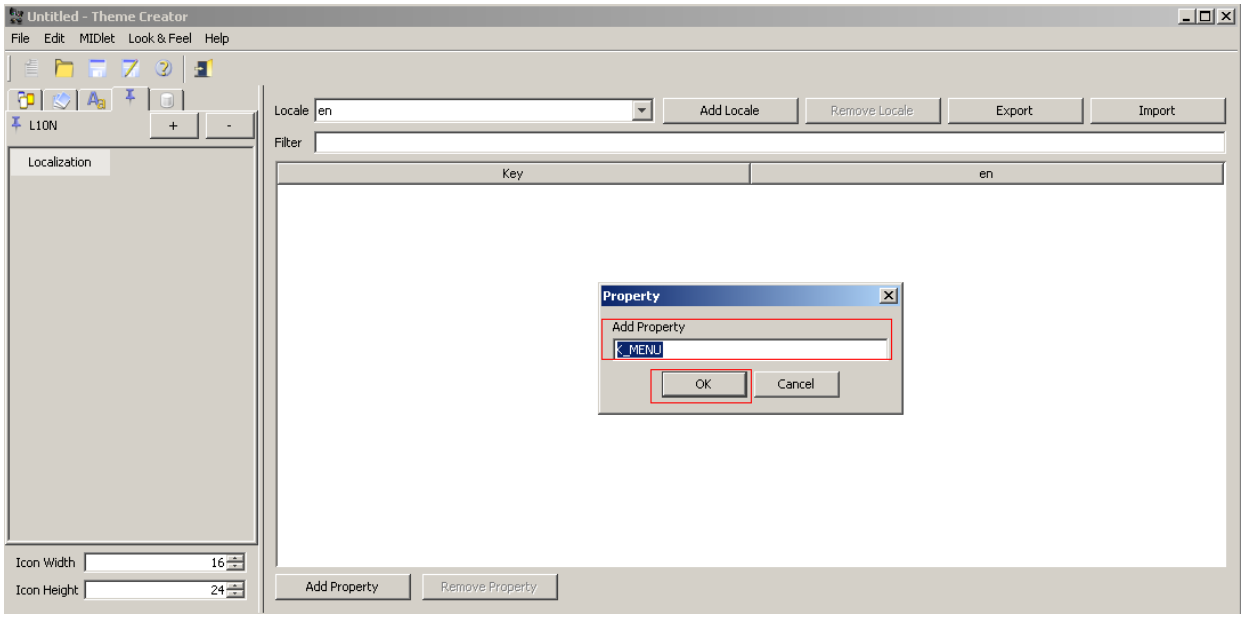
Type a name and click OK



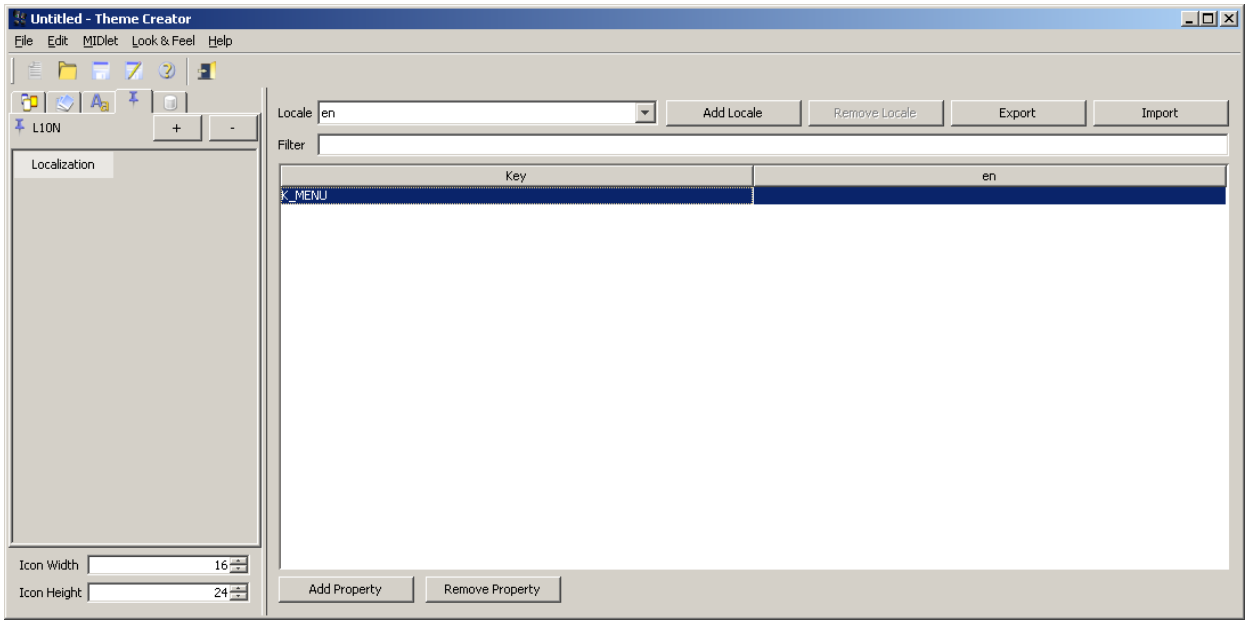
Select locale and click Add Property



Type a name of property and click OK



A property with the given name will appear inside tool



Now add the language specific value in en column.