NavScreen Windows Customization Guide

Customization Guide

Customer Support

MT6000

Doc No: CS6000-L1C-CGD-V1.0EN
Version: V1.0
Release date: 2016-12-23
Classification: internal

Keywords
Customization Guide

MediaTek Inc.

Postal address
No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

MTK support office address
No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

Internet
http://www.mediatek.com/

Document Revision History

Revision	Date	Author	Description
V1.0	2016-12-23	Shan Zhang	Initial Release

Table of Contents

Lists of Tables

Lists of Figures

1 Introduction

1.1 Purpose

NavScreen window is a new feature base on Google's default Browser on Android.It is a new view mode of Windows when user touches "Windows" option menu on Browser's main page,just like the existing list view mode.User now can switch the view mode through option menu

NavScreen window using a common view call NavScreen and a adapter called TabAdapter which is a inner class of NavScreen.This adapter has three parts:Title,Close Button and a Bitmap.

All customization code used in this document is in Browser app. And the code path is :

vendor/mediatek/proprietary/packages/apps/Browser/src/com/android/browser

1.2 Scope

The document provide the programming details of the NavScreen window.

1.3 Who Should Read This Document

This document is primarily intended for:

- Engineers with technical knowledge of the browser
- Customers who want to customize the default NavScreen window

1.4 How to Use This Manual

This segment explains how information is distributed in this document, and presents some cues and examples to simplify finding and understanding information in this document. **Error! Reference source not found.** presents an overview of the chapters and appendices in this document.

Table 1-1. Chapter Overview

#	Chapter	Contents
1	Introduction	Describes the scope and layout of this document.

1.4.1 Terms and Conventions

This document uses special terms and typographical conventions to help you easily identify various information types in this document. These cues are designed to simply finding and understanding the information this document contains.

Table 1-2. Conventions

Convention	Usage	Example
[1]	Serial number of a document in the order of appearance in the References topic	Look up Chapter 2: System Architecture in [1]
void xx(zz)	Source code	static int __stdcall cb_download_bloader_init(void *usr_arg){}
☞	Important	

2 Customization items list

Customization items list is as follows:

1. Basic operation
2. TabView size
3. Fling velocity

3 Abbreviations

Please note the abbreviations and their explanations provided in **Error! Reference source not found.**. They are used in many fundamental definitions and explanations in this document and are specific to the information that this document contains.

Table 3-1. Abbreviations

Abbreviations	Explanation
MTK	MediaTek, Asia's largest fabless IC design company.

4 Overview

This chapter first gives a brief description of the modules of the system and the relationship of the modules.

4.1 Architecture

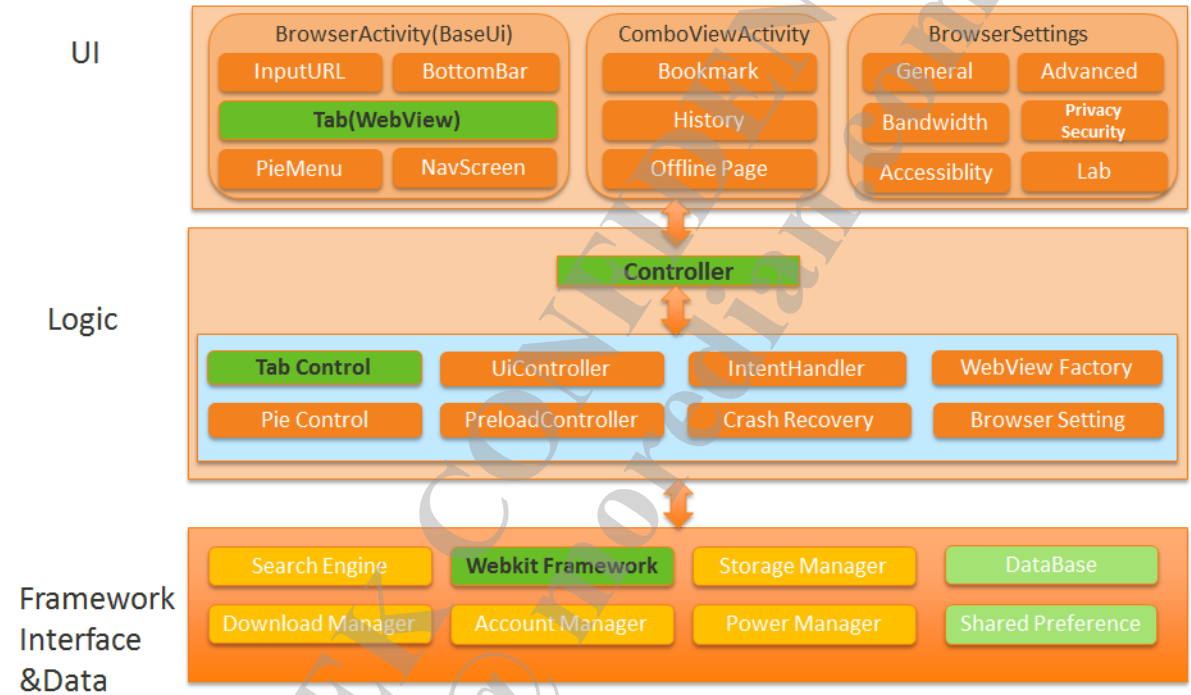


Figure 4-1. Browser Architecture

4.2 Source Code Organization

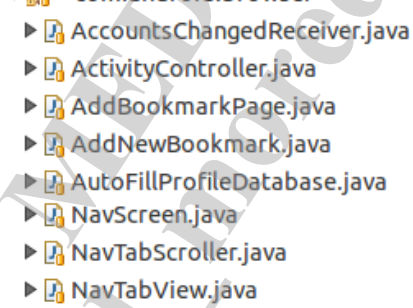


Figure 4-2. Source Code Directory Structure

The description of the directories and their subdirectories is given below:

com/android/browser	Contains the top-level source directory
com/android/browser/NavScreen	Contains the navscreen windows entity

5 Detailed Customization Procedure

5.1 Basic Operation

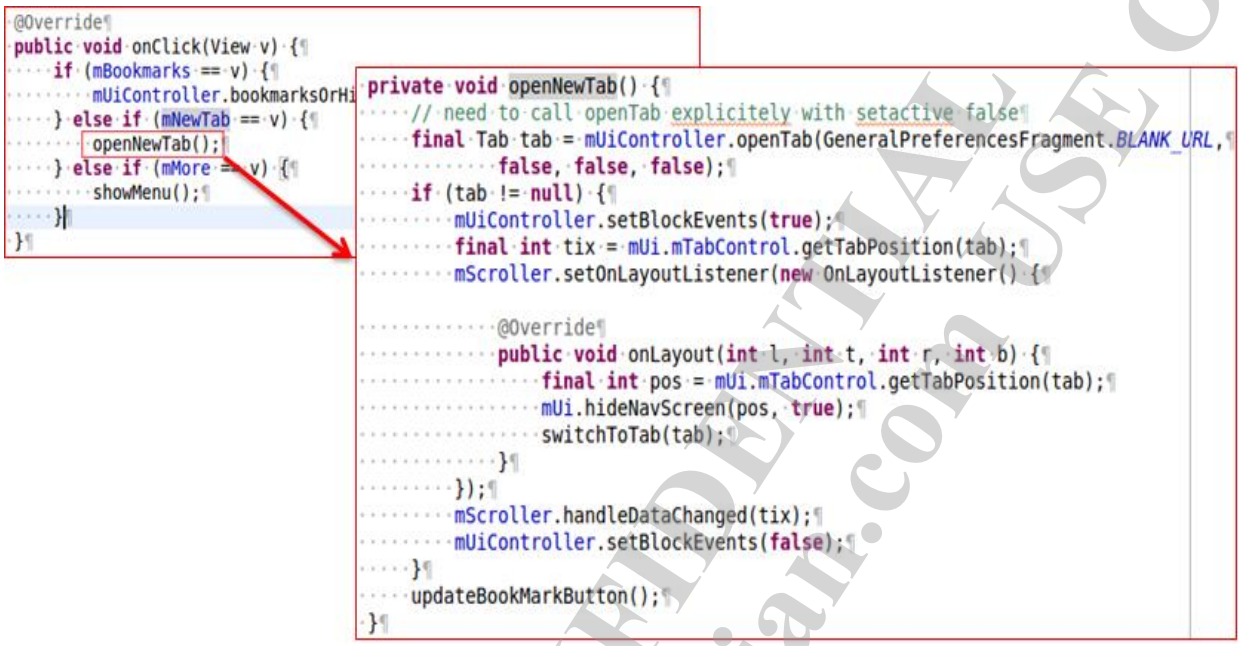
5.1.1 Title bar

In the navscreen window page,there is a title bar,you can click the “+” icon to add a new tab,or click the bookmark icon jumping to the bookmarkpage,or click the menu icon to menu items.you can see the picture as follow:

Figure 5-1.NavScreen TitleBar

1. Add a new tab

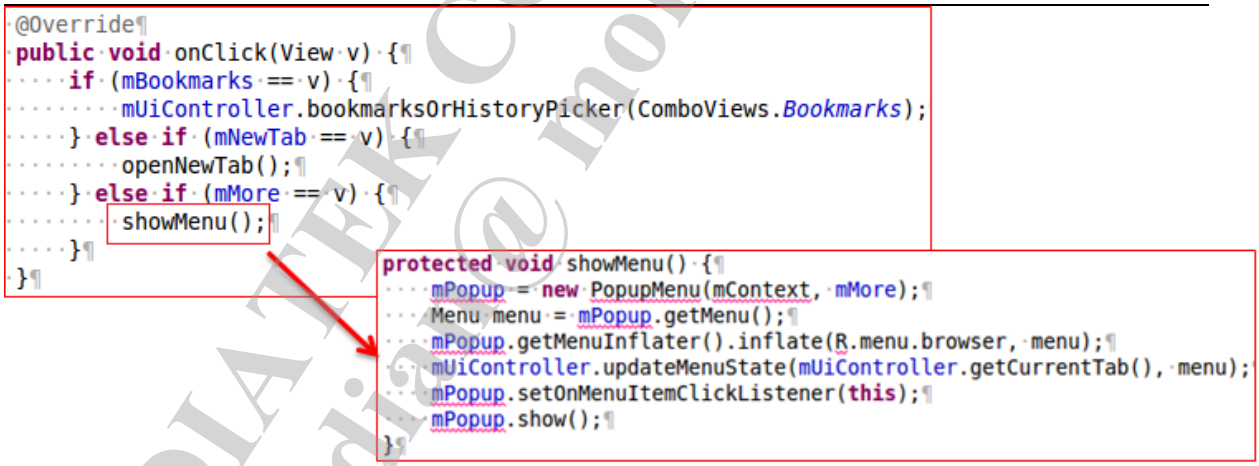
In the navscreen window page,you can click the “+” icon to add a new tab,and the related code as follow:
(src/com/android/browser/NavScreen.java::onClick)



2. Open menu items

In the navscreen window page,you can click the menu icon to popup a menu items,and the related code as follow:

(src/com/android/browser/NavScreen.java::onClick)

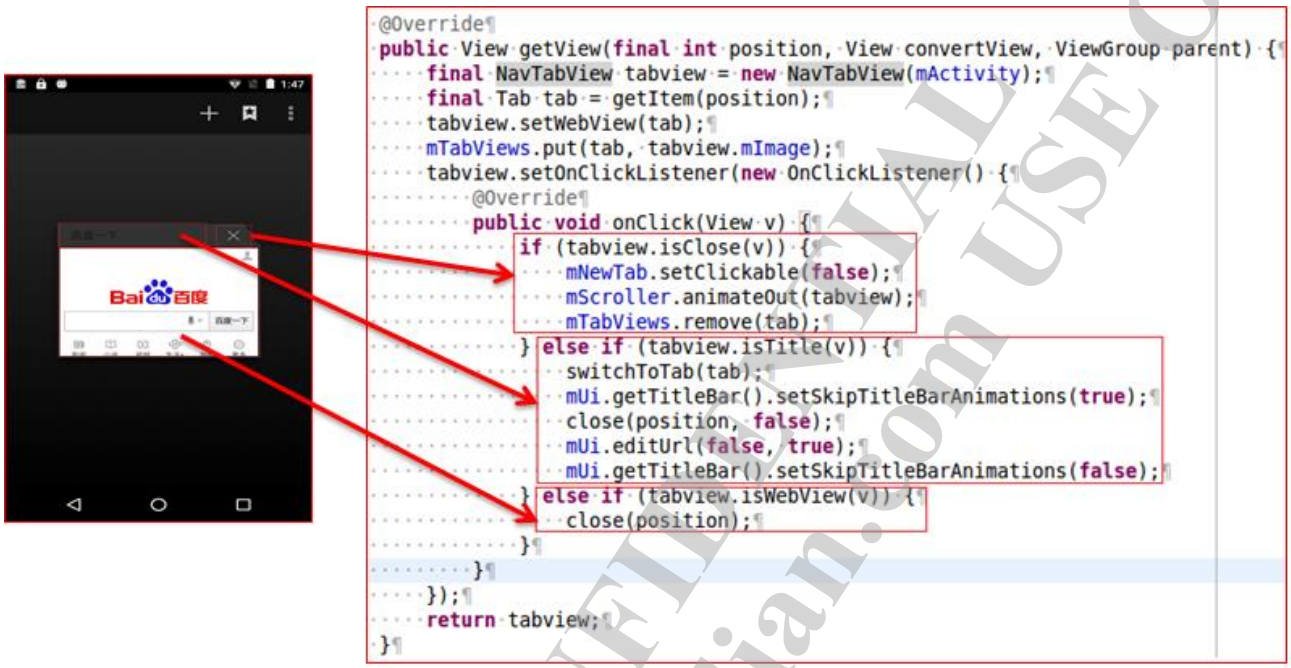


5.1.2 The thumbnail view

1. Click the thumbnail view

In the navscreen window page,you can click the “X” icon to close the current tab,or click the title and click the image to jump to the current tab,and the related code as follow:

(src/com/android/browser/NavScreen.java$TabAdapter::getView)



```
@Override
public View getView(final int position, View convertView, ViewGroup parent) {
    final NavTabView tabview = new NavTabView(mActivity);
    final Tab tab = getItem(position);
    tabview.setWebView(tab);
    mTabViews.put(tab, tabview.mImage);
    tabview.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            if (tabview.isClose(v)) {
                mNewTab.setClickable(false);
                mScroller.animateOut(tabview);
                mTabViews.remove(tab);
            } else if (tabview.isTitle(v)) {
                switchToTab(tab);
                mUi.getTitleBar().setSkipTitleBarAnimations(true);
                close(position, false);
                mUi.editUrl(false, true);
                mUi.getTitleBar().setSkipTitleBarAnimations(false);
            } else if (tabview.isWebView(v)) {
                close(position);
            }
        }
    });
    return tabview;
}
```

2. Fling the thumbnail view

In the window list,you can silde anyone window, and the window will gradient, and move ,until the window delete.If move too small,having springback function.and the related code as follow:

(src/com/android/browser/view/ScollerView.java::onTouchEvent)

```
case MotionEvent.ACTION_UP:
    final VelocityTracker vtracker = mVelocityTracker;
    vtracker.computeCurrentVelocity(1000, mMaximumVelocity);
    if (isOrthoMove(vtracker.getXVelocity(mActivePointerId),
            vtracker.getYVelocity(mActivePointerId))
            && mMinimumVelocity < Math.abs((mHorizontal ? vtracker.getXVelocity()))
                    : vtracker.getXVelocity()))) {
        onOrthoFling(mDownView, mHorizontal ? vt
                : vtracker.getXVelocity());
        break;
    }
    if (mIsOrthoDragged) {
        onOrthoDragFinished(mDownView);
        mActivePointerId = INVALID_POINTER;
        endDrag();
```

```
@Override
protected void onOrthoDragFinished(View downView) {
    if (mAnimator != null) return;
    if (mIsOrthoDragged && downView != null) {
        // offset
        float diff = mHorizontal ? downView.getTranslationY() : downView.getTranslationX();
        if (Math.abs(diff) > (mHorizontal ? downView.getHeight() : downView.getWidth()) / 2) {
            // remove it
            animateOut(downView, Math.signum(diff) * mFlingVelocity, diff);
        } else {
            // snap
            offsetVi
        }
    }
}
```

```
private void animateOut(final View v, float velocity, float start) {
    if ((v == null) || (mAnimator != null)) return;
    final int position = mContentView.indexOfChild(v);
    int target = 0;
    if (velocity < 0) {
        target = mHorizontal ? -getHeight() : -getWidth();
    } else {
        target = mHorizontal ? getHeight() : getWidth();
    }
    int distance = target - (mHorizontal ? v.getTop() : v.getLeft());
    long duration = (long) (Math.abs(distance) * 1000 / Math.abs(velocity));
    int scroll = 0;
    int translate = 0;
    int gap = mHorizontal ? v.getWidth() : v.getHeight();
    int centerView = getViewCenter(v);
    int centerScreen = getScreenCenter();
    int newpos = INVALID_POSITION;
    if (centerView < centerScreen - gap / 2) {
        // top view
        scroll = -(centerScreen - centerView + gap);
        translate = (position > 0) ? gap : 0;
        newpos = position;
```

5.2 TabView size

The followings are the thumbnail screenshots,the left one's width is 240,height is 160,and the right one's width is 300,height is 200

Figure 5-2.TabView Size Change

If you want to edit the default widow's size,you can modify the height and width in the code as follow:

(res/values/demensions.xml)

```
    <dimen name="nav_tab_width">240dip</dimen>
    <dimen name="nav_tab_height">160dip</dimen>
```

5.3 Fling Velocity

When touching the screenshot,you can drag the image sliding to left or right,the default velocity is 1500

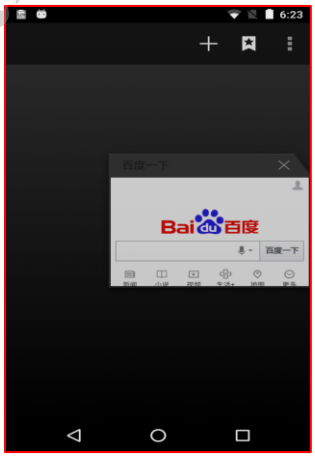


Figure 5-3.Velocity 1500

If you want to edit the default velocity,you can modify it in the code as follow:

(src/com/android/borwser/NavTabScoller.java)

```
// after drag animation velocity in pixels/sec
private static final float MIN_VELOCITY = 1500;
```