Titanium SDK 8.2.0.RC Release Note

Titanium SDK 8.2.0.RC - 6 September 2019

About this release

Titanium SDK 8.2.0 is a minor release of the SDK, addressing high-priority issues from previous releases.

As of this release, Titanium SDK 8.1.x.x will not receive updates more than six months after the release of 8.2.0 (current) (2020-03-06). Any needed fixes will be in 8.2.x or later supported releases within the 8.x branch. See Axway Appcelerator Deprecation Policy and Nominal Lifetimes documents for details.

> With the release of Titanium SDK 9.0.0, we will no longer support Node.js 8.X. Node 10.13.0 will be the new minimum supported version with SDK 9.0.0.

Notice of feature and behavior Changes

- TIMOB-27126 - iOS 13: Support dark mode
    - Added TiColor support for all color-setters
    - In iOS 13, Apple introduced support for users to adopt a system-wide Dark Mode setting where the screens, view, menus, and controls use a darker color palette. You can read more about this in the Apple Human Interface Guidelines. There are two aspects to dark mode that can be specified for your app: colors and images. To specify colors for dark mode, also known as semantic colors, first create a file called `semantic.colors.json` in the `Resources` directory for classic applications, or in the `assets` directory for Alloy applications. Then you can specify color names in the following format:

```
{
  "textColor": { // the name for your color
    "dark": {
     "color": "#ff85e2", // hex color code to be set
     "alpha": "50.0" // can be set from a range of 0-100
   },
   "light": "#ff1f1f"
  }
}
```

    - To reference these colors in your application use the Titanium.UI.fetchSemanticColor API method, this is a cross platform API that on iOS 13 and above will use the native method that checks the users system-wide setting, and in all other instances will check the Titanium.UI.semanticColorType property and return the correct color for the current setting.
    - Note: Dark Mode images are iOS only.
    - To specify dark mode images, use the `-dark` suffix on the image name. When building your app the images are set as the dark mode variant, then refer to images as normal and iOS will select the correct image dependent on the users system-wide setting.
    - For example given an image `logo.png` with `@2x` and `@3x` variants, the following dark mode images should exist:
        - logo-dark.png
        - logo-dark@2x.png
        - logo-dark@3x.png
    - And you would reference the image as before using `logo-dark.png`

New features

- TIMOB-26566 - iOS: Add ability to remove previously added motion effects
  - Added `clearMotionEffects()` method
- TIMOB-27127 - iOS 13 : Expose new APIs to support Apple login
  - Added ti.applesignin 1.1.0 module
- TIMOB-27133 - iOS 13: Add support for SF Symbols
  - Added support for SF Symbols

Expand

source

```
var win = Ti.UI.createWindow({
    backgroundColor: '#fff'
});
var triangle = Ti.UI.iOS.systemImage('drop.triangle.fill');
var forward = Ti.UI.iOS.systemImage('forward');
var backward = Ti.UI.iOS.systemImage('backward');
var folder = Ti.UI.iOS.systemImage('folder.fill');

var imageView = Ti.UI.createImageView({
    width: '150',
    height: '150',
    image: triangle,
    tintColor: 'red'
});

var button = Ti.UI.createButton({
    top: 60,
    width: 100,
    height: 50,
    //backgroundImage: folder
    image: folder
})

var stepper = Ti.UI.iOS.createStepper({
    top: 150,
    incrementImage: forward,
    decrementImage: backward
});

win.add(button);
win.add(stepper);
win.add(imageView);
win.open();
```

- TIMOB-27142 - iOS 13 : Multiple row selection in ListView
  - Added multiple row selection support in ListView and TableView

Ti.UI.ListView

Expand

source

```
var win = Ti.UI.createWindow({
    backgroundColor: '#fff'
});

var nav = Ti.UI.createNavigationWindow({
  window: win,
 });
```

```
 var items = [];
 for (var i = 0; i < 20; i++) {
    items.push({
            properties: {
                title: 'Item ' + i,
                canEdit: true,
            }
        });
  }

var list = Ti.UI.createListView({
    allowsMultipleSelectionDuringEditing: true,
    allowsMultipleSelectionInteraction: true,
    sections: [Ti.UI.createListSection({
                items: items
            })]
})

 list.addEventListener('itemclick', function(e) {
    Ti.API.info('click at index: ' + e.itemIndex);
})

 list.addEventListener("delete", function(e){
    Ti.API.info("Deleted Row Index is is: " +e.itemIndex);
    Ti.API.info("Deleted Section Index is is: " +e.sectionIndex);

 });

  list.addEventListener("itemsselected", function(e){
    Ti.API.info("Selected  Item count is: " +e.selectedItems.length);


  var dialog = Ti.UI.createAlertDialog({
    buttonNames: ['Change Color', 'Cancel'],
    message: 'Would you like to change title color of selected rows?',
  });

  dialog.addEventListener('click', function(f) {
    if (f.index === 1) {
      list.editing = false;
    } else {
        for (var i = 0; i < e.selectedItems.length; i++) {
            var rowObject = e.selectedItems[i];
            var item = rowObject.section.getItemAt(rowObject.itemIndex)
            item.properties.color = 'green';
            rowObject.section.updateItemAt(rowObject.itemIndex, item);
        }
        list.editing = false;
    }
  });
 dialog.show();

});
```

```
win.add(list);
nav.open();
```

Community credits

- VDLP - TIMOB-27207
- Christy Thomas - TIMOB-24171
- Fabian Martinez - TIMOB-27205
- Sergey Volkov - TIMOB-27234
- Hans Knöchel - TIMOB-27263, TIMOB-27367, TIMOB-27169, TIMOB-27171, TIMOB-27209

Fixed issues

- TIMOB-24171 - iOS: AccessibilityLabel or AccessibilityValue does not work.
- TIMOB-27195 - iOS: Using a commonjs module in an itemtemplate fails in a classic app
- TIMOB-27205 - Webview http redirects not working as on 7.5.1
- TIMOB-27234 - iOS: VideoPlayer natural size
- TIMOB-27263 - iOS 13: Modal windows with large titles do not honor barColor
- TIMOB-27292 - iOS: App crashes on startup on iOS 9
- TIMOB-27351 - iOS: Unit tests failing on Xcode 11 / iOS 13
- TIMOB-27360 - Watch app does not get installed on the watch with watch OS 6.0 & IOS 13
- TIMOB-27367 - iOS: Apps can be rejected when including UIWebView refs

Improvements

- TIMOB-26573 - TiAPI: Make Ti.Buffer Node-compatible with Node's Buffer type
    - Added Node.js compatible Buffer module
- TIMOB-27125 - iOS 13: Make iOS development-project compatible with Xcode 11
    - Added support for Xcode 11 and iOS 13 development environment
- TIMOB-27163 - iOS: Update Xcode project template settings and resolve warnings
    - Resolved project warnings for missing method selectors, nullability flags, and strict prototypes
- TIMOB-27169 - iOS 13: Prevent modal windows from being swiped down
    - Added support to prevent modal windows from being swiped down

Expand source

```
var window1 = Ti.UI.createWindow({
    title: "Modal Window",
    backgroundColor: 'white'
});

var win = Ti.UI.createNavigationWindow({ window: window1 });

var button1 = Ti.UI.createButton({ title: 'Open Window' });

window1.add(button1);

win.open();

var window2 = Ti.UI.createWindow({ backgroundColor: 'blue' });

var button2 = Ti.UI.createButton({ title: 'Close Window' });

window2.add(button2);

button1.addEventListener('click', function(e){
  window2.open({
    modal:true,
    forceModal: true
  });
});

button2.addEventListener('click', function(e){
  window2.close();
});
```

- TIMOB-27171 - iOS 13: Support new UITableViewStyleInsetGrouped style in list-view
  - Added support for UITableViewStyleInsetGrouped style in ListView
- TIMOB-27209 - iOS: Be able to determine dark / light mode, as well as changes on it
  - Added feature that allows for detection of dark and light mode

Expand source

```
var currentStyle = Ti.App.iOS.userInterfaceStyle;

console.log('Initial style:' + formattedUserInterfaceStyle(currentStyle));

var win = Ti.UI.createWindow({
    backgroundColor: '#fff'
});

var btn = Ti.UI.createButton({
    title: 'Check User Interface Style'
});

btn.addEventListener('click', function() {
    Ti.API.info('User Interface Style: ' +
formattedUserInterfaceStyle(currentStyle));
});

Ti.App.iOS.addEventListener('traitcollectionchange', function (event) {
    if (currentStyle !== Ti.App.iOS.userInterfaceStyle) {
        currentStyle = Ti.App.iOS.userInterfaceStyle;
        Ti.API.info('User Interface Style changed: ' +
formattedUserInterfaceStyle(currentStyle));
    }
});

win.add(btn);
win.open();

function formattedUserInterfaceStyle(style) {
    switch (style) {
        case Ti.App.iOS.USER_INTERFACE_STYLE_LIGHT: return 'Light';
        case Ti.App.iOS.USER_INTERFACE_STYLE_DARK: return 'Dark';
    }

    return 'Unspecified';
}
```

- TIMOB-27273 - iOS 13: Support new type of status bar style UIStatusBarStyleDarkContent
    - Added support for UIStatusBarStyleDarkContent

Expand source

```
var win = Ti.UI.createWindow({
    backgroundColor: 'white',
    statusBarStyle: Ti.UI.iOS.StatusBar.DARK_CONTENT
});

var btn = Ti.UI.createButton({ title: 'Trigger' });

btn.addEventListener('click', function() {
    Ti.API.info('Hello world!');
    Ti.API.info(Ti.UI.iOS.StatusBar.DARK_CONTENT);
});

win.add(btn);
win.open();
```

- TIMOB-27310 - iOS 13: Support new type of UIBlurEffectStyle constants
  - Added support for UIBlurEffectStyle
- TIMOB-27318 - iOS: Use swift 5 for apple watch template
  - Updated Swift to version 5 for Apple watch template
- TIMOB-27358 - iOS: Support Apple Developer certificates
  - Added support for generic Apple certificates

Known issues

- TIMOB-27362 - Hyperloop example with Xcode 11 throws error on pod
  - Building the Pods project with `xcodebuild` from Xcode 11 fails if `project.xcworkspace/contents.xcworkspacedata` is not present

API changes

New APIs

The following APIs are new or have expanded platform support in release 8.2.0.

API	Type	Notes
Titanium.App.iOS.USER_INTERFACE_STYLE_DARK	property	A dark interface style. (New API, supported o iPad.)
Titanium.App.iOS.USER_INTERFACE_STYLE_LIGHT	property	A light interface style. (New API, supported o iPad.)
Titanium.App.iOS.USER_INTERFACE_STYLE_UNSPECIFIED	property	An unspecified interface style. (New API, sup and iPad.)
Titanium.App.iOS.getUserInterfaceStyle	method	Access Titanium.App.iOS.userInterfaceStyle
Titanium.App.iOS.traitcollectionchange	event	Fired when the trait collection of the device c user interface style. (New API, supported on
Titanium.App.iOS.userInterfaceStyle	property	The style associated with the user interface. supported on iPhone and iPad.)
Titanium.UI.ListView.allowsMultipleSelectionInteraction	property	Allows a two-finger pan gesture to automatic table view into editing mode and start selecti API, supported on iPhone and iPad.)
Titanium.UI.ListView.getAllowsMultipleSelectionInteraction	method	Access Titanium.UI.ListView.allowsMultipleS instead.

Titanium.UI.ListView.itemsselected	event	Fired when user stops two-pan gesture inter multiple items.It is used with Titanium.UI.ListView.allowsMultipleSelectior API, supported on iPhone and iPad.)
Titanium.UI.ListView.setAllowsMultipleSelectionInteraction	method	Set the value using Titanium.UI.ListView.allowsMultipleSelectior instead.
Titanium.UI.SEMANTIC_COLOR_TYPE_DARK	property	Return the dark value from the applications c supported on Android, iPhone and iPad.)
Titanium.UI.SEMANTIC_COLOR_TYPE_LIGHT	property	Return the light value from the applications c supported on Android, iPhone and iPad.)
Titanium.UI.TableView.allowsMultipleSelectionDuringEditing	property	Determines whether multiple items of this tal selected at the same time while editing the ta supported on iPhone and iPad.)
Titanium.UI.TableView.allowsMultipleSelectionInteraction	property	Allows a two-finger pan gesture to automatic table view into editing mode and start selecti API, supported on iPhone and iPad.)
Titanium.UI.TableView.getAllowsMultipleSelectionDuringEditing	method	Access Titanium.UI.TableView.allowsMultipleSelecti instead.
Titanium.UI.TableView.getAllowsMultipleSelectionInteraction	method	Access Titanium.UI.TableView.allowsMultipleSelecti instead.
Titanium.UI.TableView.rowsselected	event	Fired when user stops two-pan gesture inter multiple rows.It is used in conjunction with Titanium.UI.TableView.allowsMultipleSelecti API, supported on iPhone and iPad.)
Titanium.UI.TableView.setAllowsMultipleSelectionDuringEditing	method	Set the value using Titanium.UI.TableView.allowsMultipleSelecti instead.
Titanium.UI.TableView.setAllowsMultipleSelectionInteraction	method	Set the value using Titanium.UI.TableView.allowsMultipleSelecti instead.
Titanium.UI.View.clearMotionEffects	method	Removes all previously added motion effects supported on iPhone and iPad.)
Titanium.UI.fetchSemanticColor	method	Fetches the correct color to be used with a U dependent on the users current dark mode s and above, or the Titanium.UI.semanticColo other instances.(New API, supported on And iPad.)
Titanium.UI.getSemanticColorType	method	Access Titanium.UI.semanticColorType inste
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_CHROME_MATERIAL	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_CHROME_MATERIAL_DARK	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_CHROME_MATERIAL_LIGHT	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_MATERIAL	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_MATERIAL_DARK	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_MATERIAL_LIGHT	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)

Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_THICK_MATERIAL	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_THICK_MATERIAL_DARK	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_THICK_MATERIAL_LIGHT	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_THIN_MATERIAL	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_THIN_MATERIAL_DARK	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_THIN_MATERIAL_LIGHT	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_ULTRA_THIN_MATERIAL	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_ULTRA_THIN_MATERIAL_DARK	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.BLUR_EFFECT_STYLE_SYSTEM_ULTRA_THIN_MATERIAL_LIGHT	property	Use with BlurView.effect to specify a blur eff supported on iPhone and iPad.)
Titanium.UI.iOS.TableViewStyle.INSET_GROUPED	property	A table view whose sections present distinct and grouped sections are inset with rounded section headers and footers do not float.(Ne on iPhone and iPad.)
Titanium.UI.iOS.systemImage	method	Get image from SF Symbols provided by Ap supported on iPhone and iPad.)
Titanium.UI.semanticColorType	property	When running on Android, iOS 10 or lower, value to return form the applications colorset supported on Android, iPhone and iPad.)
Titanium.UI.setSemanticColorType	method	Set the value using Titanium.UI.semanticCol

SDK modules versions

Android and iOS

Module	Android version	iOS version
urlSession	n/a	2.2.0
facebook	8.0.0	6.0.0
ti_coremotion	n/a	2.0.1
ti_map	4.3.1	3.1.2
ti_safaridialog	n/a	1.1.1
ti_webdialog	1.1.0	1.1.0
ti_touchid	3.0.1	2.1.4
ti_identity	2.1.0	1.0.6
ti_cloudpush	6.0.1	n/a
ti_playservices	16.1.3	n/a

CommonJS

Module	Version
ti.cloud	3.2.11

Hyperloop

Module	Version
Hyperloop	4.0.3