Oracle® Cloud

Known Issues for Oracle Visual Builder

Release 21.04.1
F38397-03
April 2021

Oracle Cloud Known Issues for Oracle Visual Builder, Release 21.04.1

F38397-03

Primary Author: Oracle Corporation

Contents

Preface

Known Issues describes known issues for Oracle Visual Builder.

Topics:

Related Resources

For more information, see these Oracle resources:

- Oracle Public Cloud

  `http://cloud.oracle.com`
- *Using Oracle Visual Builder Cloud Service*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Known Issues

This section describes issues associated with Oracle Visual Builder.

Topics:

Supported Browsers

Oracle Visual Builder Studio supports most modern HTML5–compliant browsers.

VB Studio complies with the Oracle Software Web Browser Support Policy and supports the latest version of the browser available, and in the case of IE and Safari, one previous major release. Support is provided by Oracle on all platforms that the browser vendor provides support for. For mobile device operating systems, Oracle provides support for the most recent browser delivered by the device operating system only.

The following table describes the platforms supported by the Visual Builder runtime.

Operating System	Chrome	Firefox	Microsoft Browser	Safari
Android	Supported*	Not Supported	N/A	N/A
iOS	Not Supported	Not Supported	N/A	Supported
Mac OS X	Supported	Supported	N/A	Supported
Windows	Supported	Supported	Supported	Not Supported

* Chrome for Android only, not native Android browser

JavaScript must be enabled for all browsers.

> **Note:**
>
> The Visual Builder designtime UI for building applications supports the Chrome browser running on Mac OS X and Windows. Other browsers and platforms are not supported.

> **Note:**
>
> Visual Builder runtime has deprecated the use of Internet Explorer 11. Users who try to access a staged or published Visual Builder application from Internet Explorer will now see a deprecation warning. Support for Internet Explorer 11 will be discontinued in a future release of Visual Builder in the second half of 2020.

Known Issues

Known issues associated with Oracle Visual Builder 21.04.1.

Topics:

App staging fails because of service worker download issues

Staging an application fails with status 500 and shows the following error message:

```
"IOException: Cannot download service worker from https://
static.oracle.com/cdn/vb/2104.0.1/vb-service-worker.js;
   can't read data from a response - message: Unable to parse "Content-
Type" header value: "application/javascript; charset=""
```

There's no workaround for this issue, but a fix will be available soon.

Settings for PWA-enabled mobile apps are missing

For a mobile app enabled as a PWA, the Application Name, Short Name, and Description settings don't show on the PWA tab. This occurs when the app is closed and re-opened.

To change these values, update the application's `manifest.json` file, which you can access under `mobileApps` on the Source View tab.

Camera component not working on Android Hybrid mobile apps

In Android hybrid mobile apps, the camera component won't take photos or videos.

To add this functionality to your app, use the Take Photo action in an action chain.

Copy to clipboard button doesn't copy catalog API URLs

Clicking the clipboard button doesn't copy the API URL to the clipboard in the Catalog API pane of the business object Settings tab and in the Resource API pane of a business object. You'll need to explicitly select and copy the URL.

Transform descriptions not found for dynamic services

Transforms defined for a dynamic service might not function correctly because the runtime is looking for the transforms in the wrong location.

Dynamic service connection not working when relative URL defined in OpenAPI3 document

Creating a service connection using OpenAPI3 specification and the "Dynamically Retrieve metadata" option will give a runtime error when trying to fetch/post data if the source OpenAPI3 document has a relative URL, for example:

```
 "servers":[
   {
       "url":"/api/v3"
   }
]
```

The workaround is to deselect the "Dynamically retrieve metadata" option and create a static service connection.

Message About Updating App for Future Version Compatibility on iOS

When apps are distributed through the Apple Developer Enterprise Distribution program, launching a new or an existing app on iOS 14.2 or later for the first time shows the following message:

`App app_name needs to be updated. This app will not work with future versions of iOS. The developer of this app needs to update it to improve its compatibility.`

This message follows an Apple update that relates to code signature formats used in certificates. You can safely ignore the message and click **OK** to launch your app.

This issue does not apply to apps distributed through the App Store.

Uploading signed app to Google Play Store fails with error message about app using API Level 28 where the target must be at least API 29

While uploading new or existing apps, if your app uses custom Cordova plugins that use API level 28, you'll get an "Upload failed" message when you try to upload the app to the Google Play Store. Google requires that your app uses at least API level 29 to ensure it's built on the latest APIs optimized for security and performance.

```
Upload failed.
Your app currently targets API level 28 and must target at least API level 29
to ensure it is built on the latest APIs optimized for security and performance.
Change your app's target API level to at least 29.
```

To resolve this issue, use a custom Cordova template (if you haven't been using it already). Then update the `android-targetSdkVersion` property in your application's `config.xml` to use version 29:

```
<preference name="android-targetSdkVersion" value="29" />
```

For information about how to add or update custom Cordova plug-ins in an Oracle Visual Builder app, see Add a Custom Plugin to Your Mobile Application.

Error when using a Call Module Function with the shorthand "module": "[[$functions]]"

Visual Builder throws an exception if you try to modify the `this` object. For example, writing code as follows for a Call Module Function will throw an exception:

```
PageModule.prototype.doSomething = function () {
    this.x = 1;
   }
```

Visual Builder throws an exception if you try to modify it. To avoid this issue, specify the longhand version in the JSON entry for the Call Module Function:

```
"actions": {
        "callModuleFunction1": {
          "module": "vb/action/builtin/callModuleFunctionAction",
          "parameters": {
            "module": "[[ $page.functions ]]",
            "functionName": "helloWorld"
```

Update Your Oracle SaaS Application Template Theme

If your visual application uses a theme provided by an Oracle SaaS application template, you will need to update the template when new versions are released.

New versions are typically released when a new version of Oracle JET contains changes that might affect how components used by the template are displayed. After downloading the new application template, you import it into your application as a resource and then select the new theme provided by the template in the app's Settings editor.

If you have existing visual applications that use the sample R13 SaaS LightBlue theme from an earlier version, you'll need to upgrade the theme before running the app under Oracle Visual Builder 20.10. To upgrade the theme, download the updated version of the Oracle SaaS R13 theme files (`ApplicationsCloudUI-export-4.0.0.zip`) from the vbcs-samples repository on GitHub, import the update theme resources into your application and select the updated theme in your web app's Settings editor. For details on the upgrade steps, see Upgrading the Sample R13 SaaS LightBlue Theme for Visual Builder 20.10.

Loading JSON using Text module stops Persistence Toolkit working

If you add JSON resources to your app, as shown in these code snippets, the Oracle Offline Persistence Toolkit stops working.

```
// Add resource
define([
  'text!resources2/js/test.json'
], function(

// Where resources2 is defined as:
  "requirejs": {
        "paths": {
            "resources2": "{{ location.pathname +
(window.vbInitConfig.BASE_URL_TOKEN ?
                          window.vbInitConfig.BASE_URL_TOKEN + '/' :
'') + 'resources'}}"
        }
    },
```

The following error occurs when the app tries to load the Oracle Offline Persistence Toolkit.

```
Failed to load offline handler from /ic/builder/rt/otf/2.0/webApps/foo/
version_279023731935549663/app-flow.js: Error: importScripts failed
for /ic/builder/rt/otf/2.0/
webApps/foo/version_279023731935549663/app-flow.js at /ic/
builder/rt/otf/2.0/webApps/foo/versio
n_279023731935549663/app-flow.js
https://requirejs.org/docs/errors.html#importscripts
```

Two possible workarounds exist to address this issue:

1. Wrap the JSON file in a JavaScript file and load the JavaScript file.
2. Use the Fetch API to load the JSON file in the offline handler.

On upgrade, a service authenticated with Oracle Cloud Account and using Token Relay might be migrated to connect via proxy

After migrating an application developed in Visual Builder 19.3.1, a service connection for Oracle Cloud Applications created from the catalog with authentication "Oracle Cloud Account" and the token relay checkbox selected (to allow the service to be routed directly) might be routed via the Proxy instead of going directly as intended.

If you were using token relay support with the Oracle Cloud Applications catalog, then you might need to explicitly set the connection type for the service or backend server to "Dynamic, the service supports CORS" to enable the call to the Oracle Cloud Applications service to go directly. You can set the connection type in the Settings page for the tenant or application, or explicitly on the server in the service.

2
Upgrade Policy

We strongly suggest that you upgrade and republish your apps from time to time, not only to take advantage of the new features, bug fixes, and performance enhancements that come with each new release, but also to reduce the risk of possible incompatibilities between the Visual Builder Runtime and JET libraries you built your app with in the Designer, and the Visual Builder instance to which you deployed your app in Oracle Cloud. What does this mean? Suppose you built and deployed your app using Visual Builder 18.4.1. Since then, Visual Builder has upgraded its target deployment environment several times—from 18.4.1 to 18.4.5 to 19.1.3 and so on—and your app has continued to run on those instances with no issues. However, when the 19.4.3 instance becomes available (for example), your app suddenly stumbles on an incompatibility between your 18.4.1 runtime dependencies and your 19.4.3 deployment environment—simply too much has changed between the time you built the app and the capabilities in the latest target deployment instance. To avoid this, it's a good idea to upgrade and re-publish your apps as frequently as your schedule allows. This entails opening your app in the Designer to uptake a new Runtime Version/JET library (and, optionally, taking advantage of some of the latest features), testing it with the latest Visual Builder deployment target, then republishing your app to that instance.