

## New Features Described in Oracle JET's Documentation for Release 14.0.0

Here are updates to the documentation for Oracle JET to describe new features and enhancements added to Release 14.0.0.


For other changes in this release, see the product [Release Notes](#).

### Oracle JET

Documentation updates for this release of Oracle JET.

- Oracle JET Core Pack components

This release of Oracle JET includes a new distribution of user interface components written using the VComponent API and the Oracle JET virtual DOM architecture. The `oj-c-*` DOM element prefix distinguishes these new Core Pack

components from the pre-existing components that use `oj-*`. The  badge in the Oracle JET Cookbook and the API reference doc identifies these new components. To learn more about the Core Pack components, including how to use them in an Oracle JET app that you migrate to this release, see the entry for [JET Core Pack in the API reference doc](#).

- Oracle JET CLI creates function-based VComponents by default

As of this release, the `ojet create component` command that you use to create VComponent-based web components creates function-based web components rather than class-based web components by default. That is, the following command creates a function-based VComponent named `my-vcomp`.

```
ojet create component my-vcomp --vcomponent
```

In previous releases, this command created a class-based VComponent. Now, when you want to create a class-based VComponent, you need to append the `class` value to the `--vcomponent` argument, as follows:

```
ojet create component my-vcomp --vcomponent=class
```

The `ojet create component my-vcomp --vcomponent=function` command creates a function-based VComponent. See [Hello VComponent, an Introduction](#).

- `watchInterval` property in the `oraclejetconfig.json` file  
The new `watchInterval` property with a default value of 1000 milliseconds determines the watch interval used by Oracle JET's live reload feature. Configure a different value if you want the live reload feature to poll your source files more or less frequently. See [About ojet serve Command Options and Express Middleware Functions](#).
- Oracle JET's default Webpack configuration  
A new section describes how to obtain the default Webpack configuration that Oracle JET uses when you add Webpack support to your app. It also describes how to override the default configuration options. See [Configure Oracle JET's Default Webpack Configuration \(MVVM\)](#) or [Configure Oracle JET's Default Webpack Configuration \(virtual DOM\)](#).
- The chapter describing how to use the Oracle JET Common Model and Collection API to integrate REST services into your Oracle JET has been removed from the developer's guide in this release. We recommend that you use Oracle JET's REST Data Provider APIs for this purpose. See the developer's guide for the last release of Oracle JET for more information about the [Oracle JET Common Model and Collection API](#).

## Migration Information

Note the following change that affect JET apps that you migrate to Release 14.0.0:

- JET 14.x apps will not run on Internet Explorer 11. JET Release 11.0.0 removed the ES5 libraries and polyfills that enabled JET apps in previous releases to run on Internet Explorer 11.

See the following sections for step-by-step information about migrating your app to this release of Oracle JET.

- [Migrate Alta-themed Applications from Releases Prior to 8.3.0 to Release 14.0.0](#) describes migration steps from releases prior to release 8.3.0 that allow migrating apps to the latest release.
- [Migrate to the Redwood Theme CSS](#) updated to describe how to migrate an Alta-themed app to the Redwood theme. As the topic describes, starting in release 9.0.0 it is possible to migrate to the Redwood theme.
- [Migrate Redwood-themed Apps from Releases 9.x.0 or Later to Release 14.0.0](#) describes migration steps for Oracle JET app sources from version 9.x.0 or newer to the latest release.

## Oracle JET Audit Framework

*Using and Extending the Oracle JET Audit Framework* has been updated in the following ways.

New and changed JAF audit features:

- Oracle JET releases supported by Oracle JAF

Oracle JAF does not support Oracle JET releases lower than 8.3.0. You can no longer specify 8.2.0, 8.1.0, and so on as values for the `jetVer` property in the `oraclejafconfig.json` file. To view the list of Oracle JET releases that Oracle JAF supports, enter the following command:

```
ojaf --jetlist
```

- Add custom string to runtime output

You can add a custom string for any rulepack rule and have this custom string passed through to the output when the rule to which it is applied invokes an audit issue. To add a custom string to a rule, you configure the **ruleMods** configuration property. See [Configure Audit Rule Runtime Properties](#).

- The Oracle JAF command-line interface now includes an argument (`ojaf --deplist` or `ojaf -dpl`) that displays deprecated rules. See [Audit the Application Using the Command Line](#).
- You can now configure newer versions of ECMAScript (ES12 and ES13) for the `ecmaVer` property in the `oraclejafconfig.json` file. See [Audit with Specific JET and ECMA Script Versions](#).
- JAF includes changes to existing rules:
  - **oj-html-ojevent**  
This rule now additionally tests the component metadata top-level optional property `extension/audit/ignore`, and ignores the issue for the component if the rule is declared.
  - **oj-html-wctag-packprivate**  
This rule has been elevated in severity from **major** to **blocker** to reflect its seriousness
  - The **builtinJetWcRules** and **builtinJetWcOracleRules** rule sets now recognize functional VComponents and process them accordingly
  - Wildcard support in the exclusion list for the **jetwco-component-api-controlled-changes** rule in the **builtinJetWcRules** rule pack has been expanded to allow the use of a trailing wildcard in the component name, allowing you to exclude multiple components from a specific check with a single definition.
  - The **jetwc-schema** rule in the **builtinJetWcRules** rule pack is now strict by default when auditing `component.json` files generated from VComponents. This means that any unrecognized attributes now fail the audit rather than being allowed to pass. This brings the client-side audit in step with the audits applied by the Component Exchange when uploading a component. The audit can still be configured to run in lenient mode but it is no longer the default.
  - There is a new rule group, **jet-acc**, for all accessibility rules. Members of the **jet-aria** rule group are also members of the **jet-acc** rule group.

To view all built-in rules and rule groups, run one of the following commands:

```
ojaf --grouplist
```

```
ojaf -gl
```

- JAF includes new rules:

- **jetwc-exchange-images**

Severity: major<sup>1</sup> Message ID's: jetwc-0190, 0191

This audit checks for the declaration of cover image or screen shot information in a component definition and ensures that the specified files actually exist within the packaged component. Such missing images would prevent the component from being shared via the Component Exchange

- **oj-acc-button-onclick**

Severity: blocker Message ID: jet-2180

The `ojAction` event should be used in place of `ojClick` for button-like components, because it is triggered by both mouse and keyboard.

- **jetwc-css-scoping**

Severity: blocker Message ID: jetwc-0180

Any style defined by a custom component should be scoped to only operate in the context of that component so as not to bleed through to the rest of the page and cause unexpected side effects

You can configure exceptions as described in [Audit JET Custom Web Component Projects](#).

- **jetwc-property-translatable**

Severity: minor Message ID: jetwc-0200

To promote consistency amongst a set of components this audit will verify that properties or sub-properties with a given name e.g. `pageTitle` have the `translatable` flag set. This flag is primarily used in Visual Builder to guide users to set these properties to resource bundle references rather than hardcoded strings

- **oj-html-no-style**

Severity: major Message ID: jet-0505 HTML `<style>` sections or `'style'` attributes are not permitted.

The new configurable rule **oj-html-no-style** detects use of `<style>` sections, and optionally in-line `style` attributes. This rule is completely compatible with the existing rule **oj-html-no-style-notag** (which is now deprecated, and will be removed in a future major release). If both rules are enabled, the predecessor rule will disable itself and defer to the new rule. The rule is configurable, and both `<style>` section and `style` attribute detection can be independently enabled or disabled. Like its predecessor, the new rule is disabled by default, and also defers to the **csp-html-style-unsafe-inline** rule, if enabled, since the latter rule performs more stringent checks.

Changes for rule writers:

- To avoid future naming collisions, rule options for user custom rules defined in the **rules.json** manifest, (or created/updated through the **ruleMods** config for any rules) may no longer appear at the top level. All custom rule options should now

be created under **customOpts** rule property. The limitation on option property names to avoid name collisions is no longer relevant. For compatibility, no immediate action is required because all top-level rule options unknown to JAF are currently copied into **customOpts**, and the originals are still left at the top-level. That said, it is recommended that you make the following changes to use **customOpts** because the top-level placement is deprecated and will not be permitted in a future major release of JAF:

1. In a custom rule's registration listener, direct reference to `regCtx.ruleOpts` should be replaced with `regCtx.ruleOpts.customOpts`.
2. Access to custom options through `packLib` should be modified because the new method `packLib.getRuleCustomOptions()` is available. Alternatively, the original `packLib.getRuleOptions()` can still be used, but any direct reference to a custom rule option should be changed from `packLib.getRuleOptions().my_option` to `packLib.getRuleOptions().customOpts.my_option`.

See Get Started Writing Custom Audit Rules.

- The `metaLib` utility library has new methods:
  - **isRuleIgnored()** tests component metadata top-level optional property (**extension/audit/ignore**) for the declaration of a rule name to be ignored.
  - **getTagMetadata()** returns the full `component.json` metadata as an object for the specified user-written component.
  - **getTagPropMetadata()** can also be used to retrieve the metadata for a component property/sub-property.

See `MetaLib: JET Metadata Access Functions`.

- A new constant, **PROPERTY\_DEF**, has been added to the set returned by **AstUtils.getNodeTypes()** to represent the **PropertyDefinition** AST node type that can be discovered by the TypeScript AST parsing libraries. See `Listener Types for JavaScript/TypeScript Rules`.
- A new listener type, **css-var**, has been added to allow CSS rules to listen for CSS variables. See `Listener Types for CSS Rules`.
- A new registered listener type, **attrepr-\$props**, has been added. This listens for attribute expressions that contain reference(s) to **\$properties**. Prior to the introduction of **attrepr-\$props**, a rule needed to listen for **attrepr** and then, after invocation, check if the expression contained references to **\$properties**. See `Listener Types for HTML and JSON Rules`.

## 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 Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

---

## Oracle JavaScript Extension Toolkit (Oracle JET) What's New in Oracle JET, 14.0.0

F72314-02

Copyright © 2021, 2023, Oracle and/or its affiliates

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

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, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

Intel and Intel Inside 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, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.