

Oracle® Cloud

Administering Oracle Visual Builder in Oracle Integration 3



24.10
F92646-09
February 2025

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Cloud Administering Oracle Visual Builder in Oracle Integration 3, 24.10

F92646-09

Copyright © 2023, 2025, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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, MySQL, and NetSuite 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.

Contents

Preface

| | |
|-----------------------------|---|
| Audience | v |
| Documentation Accessibility | v |
| Diversity and Inclusion | v |
| Related Resources | v |
| Conventions | v |

1 Create a Visual Builder Instance

| | |
|---|-----|
| Administer Your Oracle Integration Instances | 1-1 |
| Enable Visual Builder in Oracle Integration | 1-2 |
| Set the IAM Policy for Managing the Visual Builder Instance | 1-3 |

2 View and Manage the Visual Builder Instance

| | |
|---|-----|
| Manage Visual Builder Tags | 2-2 |
| Create and Configure a Custom Endpoint for Your Visual Builder Instance | 2-3 |
| Map Your Custom Endpoint with Your Visual Builder Instance | 2-6 |

3 Upgrade from Oracle Integration Generation 2 to Oracle Integration 3

4 Configure Tenant Settings

| | |
|---|------|
| Manage Applications in the Service Instance | 4-1 |
| Manage Applications Created on the Instance | 4-3 |
| Access Tenant Settings | 4-5 |
| Choose Your Instance's Update Window | 4-6 |
| Configure Security Options for Applications | 4-7 |
| Assign Roles for Users to Access an Application | 4-9 |
| Set Page Messages for Access Denied Errors | 4-9 |
| Allow Other Domains Access to Services | 4-10 |
| Allow Your Instance to Access Services | 4-11 |
| Inspect Database Usage | 4-13 |

| | |
|---|------|
| Switch to Your Own Oracle DB Instance | 4-14 |
| Switch From One ATP Database to Another | 4-20 |
| Make Schemas in an Oracle DB Instance Available to Applications | 4-21 |
| Update Your ATP Wallet and Reset an Expired Password | 4-22 |
| Add a Connection to Integration Applications | 4-23 |
| Add a Connection to Oracle Cloud Applications | 4-24 |
| Add a Connection to Process Automation | 4-26 |
| Add a Connection to Process Cloud Service | 4-26 |
| Add a Connection to a Custom Backend | 4-27 |
| Create a Child Backend | 4-28 |
| Edit Authentication for a Backend Service | 4-30 |
| Manage Self-signed Certificates | 4-33 |
| Manage Your Component Exchange | 4-35 |
| What is a Component Exchange? | 4-35 |
| About Component Exchanges Hosted in VB Studio Projects | 4-36 |
| Add a Connection to a Component Exchange | 4-38 |
| Configure Support for a Custom Domain | 4-38 |

Preface

This guide describes tasks for administrators of Visual Builder.

Audience

This guide is intended for administrators who will set up and configure the service.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://support.oracle.com/portal/> or visit [Oracle Accessibility Learning and Support](#) if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- Oracle Public Cloud
<http://cloud.oracle.com>
- About Visual Builder in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*
- Welcome to Oracle Integration 3 in *Getting Started with Oracle Integration 3*

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. |
| <i>italic</i> | 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

Create a Visual Builder Instance

You use the Oracle Cloud Infrastructure Console to manage Visual Builder instances provisioned in Oracle Integration, for example, to add instances.

 **Note:**

If your Visual Builder instance was not provisioned in Oracle Integration, you should refer to *Get Started with Oracle Visual Builder Generation 2* in *Administering Oracle Visual Builder Generation 2* to create and configure Visual Builder instances instead of this guide.

Administer Your Oracle Integration Instances

A Visual Builder instance can be enabled as one of the features in your Oracle Integration subscription. An administrator performs the lifecycle management tasks for Oracle Integration and Visual Builder instances in the Oracle Cloud Infrastructure Console. Depending on your subscription and environment, some options for administering instances might not be available.

The following table describes some Oracle Integration instance management tasks that administrators might need to perform when enabling a Visual Builder instance.

| Task | Description |
|---|---|
| Create an Oracle Integration instance | An administrator can provision new Oracle Integration instances. See <i>Create an Oracle Integration Instance</i> in <i>Provisioning and Administering Oracle Integration 3</i> . |
| View the Oracle Integration instance info | The Oracle Cloud Infrastructure Console provides details on all of your instances. See <i>View Instance Details</i> in <i>Provisioning and Administering Oracle Integration 3</i> . |
| Add users and assign roles | Visual Builder users need to be added to the identity domain of your Oracle Integration instance and assigned at least one of the pre-defined roles that are permitted to access the Visual Builder instance. See <i>Manage Access and Assign Roles</i> in <i>Provisioning and Administering Oracle Integration 3</i> . |
| Start and stop Oracle Integration instances | You can stop instances to free up compute resources used by the instance's virtual machines. See <i>Stop and Start an Oracle Integration Instance</i> in <i>Provisioning and Administering Oracle Integration 3</i> . |

| Task | Description |
|----------------------------------|---|
| Enable a Visual Builder instance | You enable a Visual Builder instance from the Oracle Integration Instance Details page. See Enable Visual Builder in Oracle Integration |

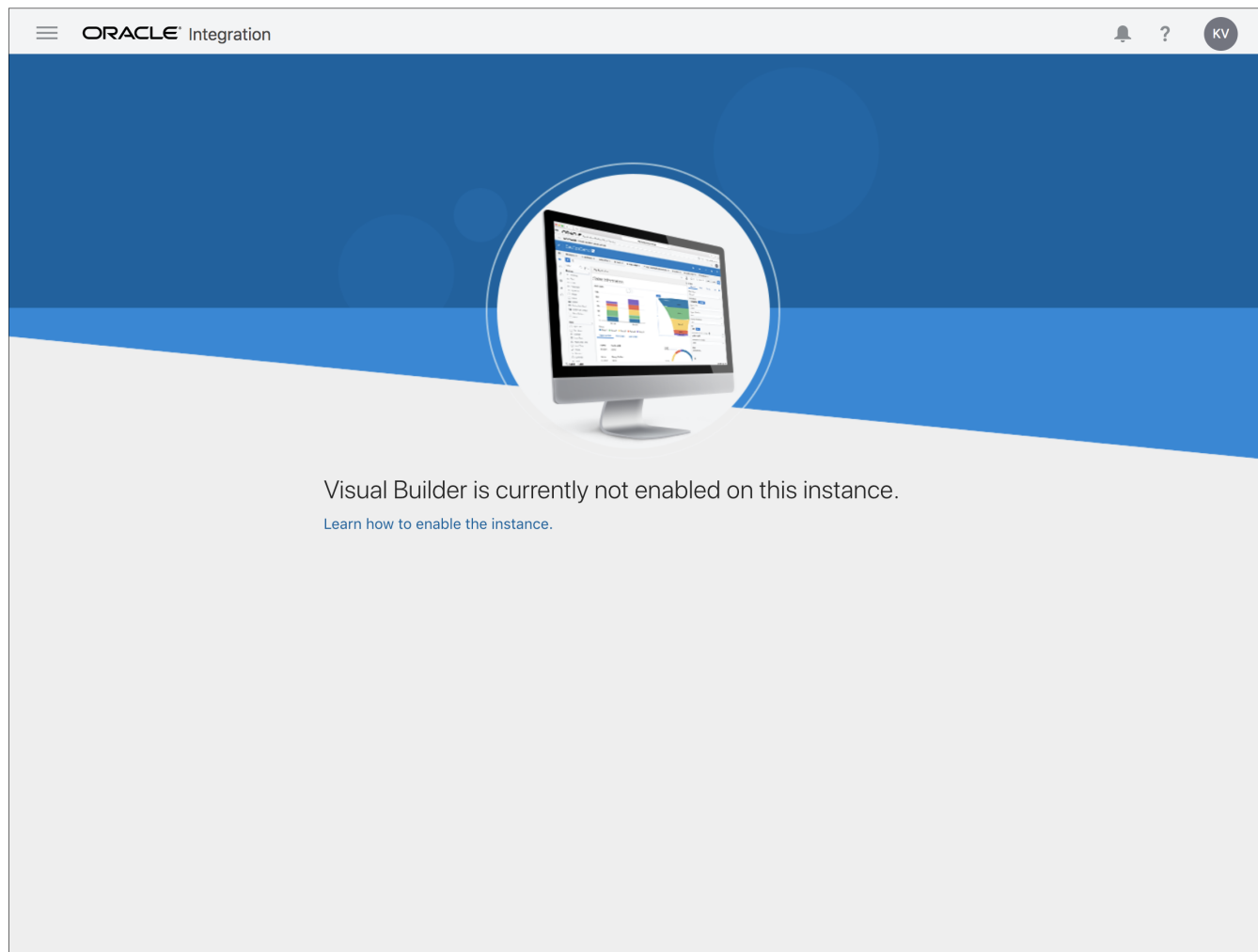
After a Visual Builder instance is enable, administrators can perform lifecycle management tasks for the instance in the Visual Builder instance's details page in the Oracle Cloud Infrastructure Console. See [View and Manage the Visual Builder Instance](#).

Additional global settings for applications created in an instance are set in the Tenant Settings page that administrators can [access directly from Visual Builder](#). See [Configure Tenant Settings](#).

Enable Visual Builder in Oracle Integration

To begin using Visual Builder in Oracle Integration, it must first be enabled for the Oracle Integration instance in the Oracle Cloud Infrastructure Console. Enabling Visual Builder is a one time action and cannot be undone.

If you select Visual Builder from the navigation pane and it's not yet enabled for Oracle Integration, the following message appears:



Important points about enabling Visual Builder:

- Enablement applies to Oracle Integration 3. A Visual Builder link is available in the side navigation pane. When you select the link, instructions for enabling Visual Builder are displayed if it is not yet enabled.
- To enable Visual Builder for an Oracle Integration instance, you must have Oracle Cloud Infrastructure manage access to the instance. See *Manage Access and Assign Roles in Provisioning and Administering Oracle Integration 3*.

When setting the policies for the VB instance, you must create a policy to allow the OIC administrator group to manage the VB instance. See [Set the IAM Policy for Managing the Visual Builder Instance](#).

- You must enable Visual Builder on each Oracle Integration instance on which you want to use it. Current Oracle Integration 3 users who have used Visual Builder in the past will be automatically enabled.

To enable Visual Builder:

1. Select your instance in the Oracle Cloud Infrastructure Console.

The Integration Instance Details page is displayed.

2. Click the **Enable** link for Visual Builder on the Integration Instance Information tab.

3. When prompted, click **Enable** to confirm you want to enable Visual Builder.

If you haven't yet set the required policies for the VB instance, you can use the **OCI Policy** link in the dialog box to open the Policies tab in Oracle Cloud Infrastructure Identity and Access Management (IAM).

After you click Enable, the Oracle Integration icon turns orange and its status changes to Updating. Enablement can take 15-30 minutes.

Once complete, the Oracle Integration icon changes back to green with an Active status, and Visual Builder shows as Enabled.

4. After your Visual Builder instance is enabled, you can:

- [View and Manage the Visual Builder Instance](#)
- [Configure Tenant Settings](#)
- Open Visual Builder:
 - a. Click **Service Console** from the buttons along the top of the Integration Instance Details tab to open Oracle Integration.
 - b. Select **Visual Builder** in the left navigation pane.

Set the IAM Policy for Managing the Visual Builder Instance

To enable Visual Builder for an Oracle Integration instance, you need to create a policy that will allow the OIC administrator group to manage the Visual Builder instance.

You'll need to open Oracle Cloud Infrastructure Identity and Access Management (IAM) to create the policy. For details about how to create policies in IAM, see *Differences Between Tenancies With and Without Identity Domains and About IAM Policies for Oracle Integration in Provisioning and Administering Oracle Integration 3*.

To create the policy:

1. Open the navigation menu and click **Identity & Security**. Under **Identity**, click **Policies**.

2. Click **Create Policy**.
3. In the Create Policy window, enter a name and description.
4. In the **Policy Builder**, select **Show manual editor** and enter the required policy statements.

Syntax:

```
allow group <admin_group> to manage visualbuilder-instance in compartment  
<compartmentId>
```

Example: allow group domain_admins to manage visualbuilder-instance in
compartment oic-dev-comp

This policy statement allows the OIC admin group domain_admins to manage the Visual
Builder instance visualbuilder-instance in compartment oic-dev-comp.


5. Click **Create**.

2



View and Manage the Visual Builder Instance

After enabling the Visual Builder instance, you can view its details and manage it from the OCI Console.

To open the Visual Builder instance in the OCI Console:

1. Open the OCI Console.
2. In the upper-left corner, click **Navigation Menu** .
3. Select **Developer Services** in the menu, then select **Integration** under **Application Integration** in the list of services.
4. If needed, select a compartment in the **Compartment** drop-down list.
5. In the table, click your Oracle Integration 3 instance to open the instance's details page. If no Visual Builder instance has been created, Visual Builder will be marked as "Not Enabled" in the information tab. You can click **Enable** to [create an instance](#).
6. Click **Associated services** under **Resources**.
7. Click the link for the Visual Builder instance in the Associated Services table.

The OCI Console has tools for administering your instance. Some options are not available for Visual Builder instances provisioned in Oracle Integration 3.

| To do this ... | Do this: |
|------------------------------|---|
| Add or edit tags | See Manage Visual Builder Tags . |
| Download activity logs | In the Activities section, click Action  and select Download Logs . |
| Download activity error logs | In the Activities section, click Action  and select Download Errors . |
| Change the node count | Changing the node count is not supported in Oracle Integration 3. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them. |
| Create a custom endpoint | See Configure a Custom Endpoint for an Instance . |
| Delete the instance | Once enabled in Oracle Integration 3, you cannot delete a Visual Builder instance. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them. |
| Start or stop the instance | Once enabled in Oracle Integration 3, you cannot stop or start a Visual Builder instance. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them. |



| To do this ... | Do this: |
|--|--|
| Move the instance to another compartment | Moving an instance to another compartment is not supported in Oracle Integration 3. In some cases you might see tools in the OCI Console to perform this task, however, Oracle recommends you do not use them. |

Some Visual Builder instance management tasks can only be performed in the Oracle Integration 3 instance's details page in the OCI Console. See [Administer Your Oracle Integration Instances](#).

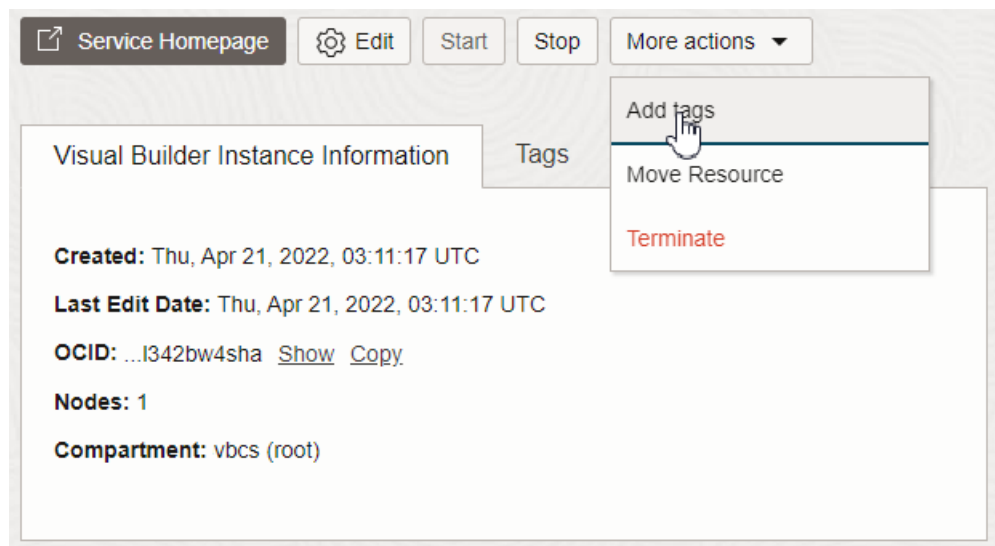
Manage Visual Builder Tags

OCI tags enable you to tag your OCI resources, such as your Visual Builder instance, and help you organize resources based on your business needs. You can manage Visual Builder tags from the Visual Builder Instance Details page.

If you're new to OCI tags, see [Tagging Overview](#).

| Action | How To |
|--------------|---|
| Add a tag | <ol style="list-style-type: none"> 1. On the Visual Builder Instance Details page, click Add Tags in the More Actions menu. 2. In the Add One Or More Tags To This Resource dialog box, enter the tag's namespace, key, and value. 3. Click Add Tags. |
| Edit a tag | <ol style="list-style-type: none"> 1. On the Visual Builder Instance Details page, click the Tags tab. 2. To edit a tag, click its Edit  icon. 3. In the Edit Tag dialog box, edit the tag and click Save. |
| Remove a tag | <ol style="list-style-type: none"> 1. On the Visual Builder Instance Details page, click the Tags tab. 2. To edit a tag, click its Edit  icon. 3. In the Edit Tag dialog box, click Remove Tag. |

You can see the Add Tags item in the More Actions menu in this image.



Create and Configure a Custom Endpoint for Your Visual Builder Instance

You can map a custom endpoint to a Visual Builder instance and use it to access the instance instead of the original URL generated in the OCI Console.

You can create a custom hostname with your chosen domain (for example, `mycustom.example.org`) for your Visual Builder instance.

Note:

Associating a custom endpoint with your Visual Builder instance doesn't affect the original instance URL. You'll be able to access your instance using the custom endpoint URL as well as the original instance URL.

Perform the following configuration tasks to successfully use a custom endpoint URL for your Visual Builder instance:

Note:

These instructions assume you have direct access to a Visual Builder instance. If you are using the instance behind a WAF or an API Gateway, skip the certificate-related instructions provided here (from Step 3 onwards) and follow the WAF or API gateway certificate instructions instead. See [WAF Certificates](#) or [Setting Up Custom Domains and TLS Certificates for API Gateways](#)

1. Choose a custom hostname for your instance and register it at a DNS provider.
2. Obtain an SSL certificate from a certificate authority (CA) for your hostname.

3. In your OCI tenancy, select a compartment and create an OCI Vault to store the certificate. See [Working with Compartments](#), [Overview of Vault](#), and [Create a New Vault](#).

 **Note:**

If you are using the instance behind a WAF, you don't need to store your certificate in a vault. You can skip this step and the next step (Step 4).

4. Store the certificate as a secret in the OCI Vault. See [Create a New Secret](#).

Use the following certificate format:

```
{
  "key": "-----BEGIN PRIVATE KEY-----\n...\n-----END PRIVATE KEY-----\n",
  "cert": "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n",
  "intermediates": [
    "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n",
    "-----BEGIN CERTIFICATE-----\n...\n-----END CERTIFICATE-----\n"
  ],
  "passphrase": "<private key password if encrypted key is provided>"
}
```

 **Note:**

- To avoid manual errors, you can also convert your PEM certificate into a single line containing "\n", as expected, with the following `awk` commands. For the leaf certificate:

```
awk -v RS= '{gsub(/\n+/, "\n")}'1' <cert_pem_file>
```

For each intermediate/root certificate:

```
awk -v RS= '{gsub(/\n+/, "\n")}'1'
<each_intermediate_cert_pem_file>
```

For the private key:

```
awk -v RS= '{gsub(/\n+/, "\n")}'1' <private_key_pem_file>
```

- The **latest** version of the secret is used when you associate a custom endpoint with your instance either through the create instance or edit instance operation. For information on secret versions, see [Secret Versions and Rotation States](#).
- If you use a hostname certificate whose certificate authority (CA) is not in the Visual Builder trust store, you must also upload the certificate to your Visual Builder instance; otherwise, an exception is thrown in the scenarios the instance calls itself.

Note the following certificate requirements:

- If the certificate has more than one root/intermediate certificate, each certificate must be specified as a separate element in an `intermediates` array.
- Always ensure that the final root CA is specified as the last element in the array. For example, if there are three intermediate certificates for the leaf certificate, the certificate that issued the leaf certificate should go as the `intermediates[0]` element, the certificate that issued the `intermediates[0]` certificate should go in the `intermediates[1]` element, and the certificate that issued the `intermediates[1]` certificate should go in the `intermediates[2]` element.
- The `passphrase` attribute is optional and does not need to be specified if the private key is not encrypted.
- If using an encrypted private key, the following format is required (PKCS1 is supported):

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
-----END RSA PRIVATE KEY-----
```

A JSON file with an encrypted private key looks as follows:

```
{
  "key": "-----BEGIN RSA PRIVATE KEY-----\nProc-Type:
4,ENCRYPTED\n...\n-----END RSA PRIVATE KEY-----",
  ..
  ..
  "passphrase": "<passphrase to decrypt the key>"
}
```

A JSON file with an unencrypted private key looks as follows:

```
{
  "key": "-----BEGIN RSA PRIVATE KEY-----
\nvRXUK08v31bw2rnDLw+vjuX2i8ujHWs\n...\n-----END RSA PRIVATE KEY-----",
  ..
  ..
}
```

- If your private key is in PKCS8 format, you must convert it to PKCS1 format:

```
openssl rsa -in <input_pkcs8_encrypted_private_key> -out
<converted_encrypted_private_key_file_name> -aes256
```

5. Create an Identity and Access Management (IAM) policy to:

- Allow the Visual Builder service to read the version and contents of the secret.

Here's the policy syntax for a Visual Builder service:

```
allow group <group-name> to read secret-bundle in compartment <secrets-
compartment>
```

Here's an example:

```
allow group VBInstanceAdmins to read secret-bundle in compartment
MySecretCompartment
```

If the VB instance is NOT in a default domain, then you need to include the domain prefix in front of the group name.

Here's an example:

```
Allow group mydomain/VBInstanceAdmins to read secret-bundle in compartment MySecretCompartment
```

- b. Allow the admin group to access the secret, key, and vault (or create a new secret, key, and vault), while creating or updating a Visual Builder instance with a custom endpoint.

Here's the policy syntax:

```
allow group <group-name> to manage secrets in compartment <secrets-compartment>
```

```
allow group <group-name> to manage keys in compartment <secrets-compartment>
```

```
allow group <group-name> to manage vaults in compartment <secrets-compartment>
```

and here are examples:

```
Allow group VBInstanceAdmins to manage secrets in compartment MySecretCompartment
```

```
Allow group VBInstanceAdmins to manage keys in compartment MySecretCompartment
```

```
Allow group VBInstanceAdmins to manage vaults in compartment MySecretCompartment
```

If the VB instance is NOT in a default domain, then you need to include the domain prefix in front of the group name.

Here are examples:

```
Allow group mydomain/VBInstanceAdmins to manage secrets in compartment MySecretCompartment
```

```
Allow group mydomain/VBInstanceAdmins to manage keys in compartment MySecretCompartment
```

```
Allow group mydomain/VBInstanceAdmins to manage vaults in compartment MySecretCompartment
```

Note that you should specify the resource to return in `<resource-type>`, as described in [Details for the Vault Service](#).

6. Create a Visual Builder instance and map the custom endpoint details to it. See [Enable Visual Builder in Oracle Integration](#) and [Map Your Custom Endpoint with Your Visual Builder Instance](#).
7. Finally, update the custom endpoint DNS record to the original instance hostname. As a best practice, update the CNAME of the DNS record with the hostname of the original instance endpoint.

Map Your Custom Endpoint with Your Visual Builder Instance

To map a custom endpoint to a Visual Builder instance, get the custom hostname for the Visual Builder instance and the OCI vault's compartment name, vault's name, and the secret key where you've uploaded the SSL certificate.

Before you begin, create a custom hostname and upload its SSL certificate to an OCI Vault, as described in [Create and Configure a Custom Endpoint for Your Visual Builder Instance](#).

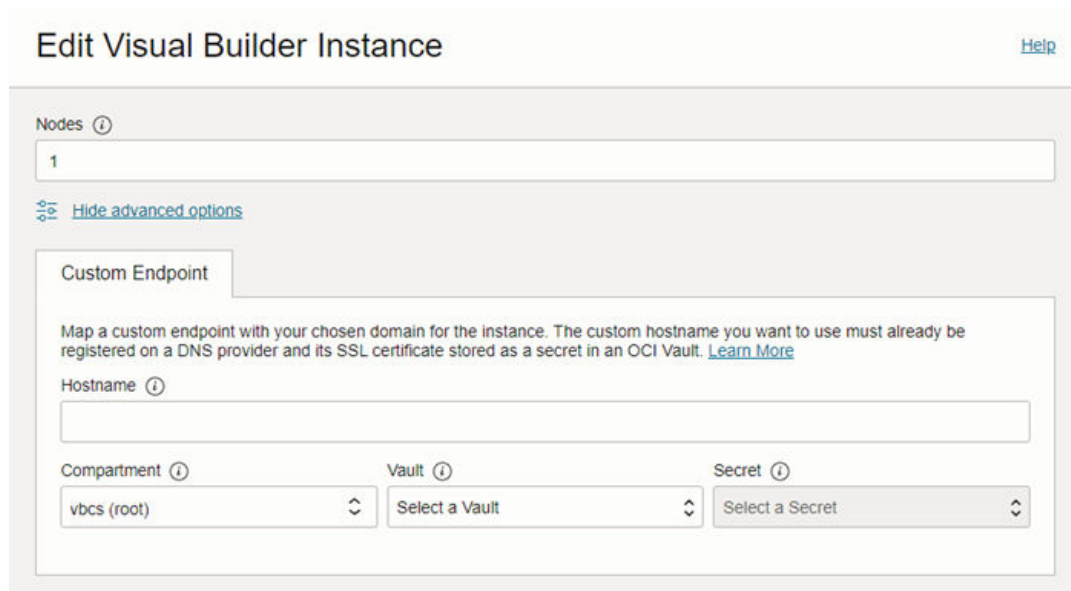
1. Open the VB instance's details page.

2. On the Visual Builder Instance Details page, click **Edit**.

The Edit Visual Builder Instance dialog is displayed. The dialog displays a Nodes field and an Advanced Options pane for adding or updating a custom endpoint. The Advanced Options pane is collapsed by default.

3. Click **Show Advanced Options**.

The Advanced Options pane contains a Custom Endpoint tab with fields for adding or updating a custom endpoint for the instance.



4. Supply the custom endpoint details.

| Field | Description |
|--------------------|---|
| Hostname | Enter the custom hostname chosen for the instance. The custom hostname you want to map to the instance must already be registered on a DNS provider and its SSL certificate stored as a secret in an OCI Vault. |
| Certificate | Provide the location of the hostname's certificate in your OCI tenancy. <ul style="list-style-type: none"> • Compartment: Select the OCI compartment that contains your certificate vault. • Vault: Select the vault that contains the hostname's certificate. • Secret: Select the secret corresponding to the hostname's certificate. |

 **Note:**

You can also update or replace a custom endpoint that was previously associated with the instance. You can modify the hostname as well as the certificate details. However, to update the certificate details, you must have access permissions to the vault containing the required certificate.

After you've mapped the instance, update the custom endpoint DNS record to the original instance hostname. As a best practice, update the CNAME of the DNS record with the hostname of the original instance endpoint.

3

Upgrade from Oracle Integration Generation 2 to Oracle Integration 3

Oracle is in the process of upgrading Oracle Integration Generation 2 to Oracle Integration 3. The upgrade is available at no extra cost.

There are some Visual Builder tasks you need to perform before and after your Oracle Integration instance is upgraded to Generation 3. For the full details on upgrading Oracle Integration, see [Upgrade from Oracle Integration Generation 2 to Oracle Integration 3](#) in *Provisioning and Administering Oracle Integration 3*.

Oracle performs the majority of the work for upgrading Oracle Integration Generation 2 instances for you, including upgrading your eligible Visual Builder instances.

For a list of known issues in Visual Builder after your Oracle Integration instance is upgraded to Generation 3, see [Known Issues for Live/Staged Apps Post-Upgrade to Oracle Integration 3](#) in *Known Issues for Oracle Integration 3*.

Upgrade Readiness Checks

Oracle periodically performs some prechecks to determine your upgrade readiness so that your upgrade runs smoothly. If the prechecks don't pass, you may need to perform tasks to correct the issues. You can see the precheck status of your Oracle Integration Generation 2 instance, or run a precheck again, in the instance's Settings page. For more, see [Check Upgrade Readiness and Correct Precheck Issues](#) in *Provisioning and Administering Oracle Integration 3*.

The following prechecks are performed for Visual Builder instances in Oracle Integration Generation 2:

- Is a custom endpoint defined in the Visual Builder instance?
The instance cannot be upgraded to Oracle Integration 3 if a custom endpoint is defined in the Visual Builder instance. To upgrade the instance, wait until Oracle starts upgrades for custom endpoints, or, if you want to move forward with the upgrade now, delete the custom endpoint from your Visual Builder instance.

After the instance is upgraded to Oracle Integration 3, you can [define a new custom endpoint in the Visual Builder instance](#).

Note:

A custom endpoint defined in a Visual Builder instance is not the same as a custom endpoint created in Oracle Integration.

- Does Visual Builder use an Oracle database instance?
If you are using your own Oracle database instance (BYODB) with your Visual Builder instance, the instance can be upgraded to Oracle Integration 3. However, Autonomous Transaction Processing (ATP) must be up and running during the upgrade. Make sure you complete the additional tasks described in [Prepare Visual Builder for the Upgrade](#) below.

Prepare Visual Builder for the Upgrade

Examine your applications, especially your live and staged applications, to check whether they might break due to any of the Known Issues for Live/Staged Apps Post-Upgrade to Oracle Integration 3. For these cases, follow the relevant instructions in *Tasks to Complete After the Upgrade* below.

- Confirm that you have upgraded your applications to the supported VB runtime and JET versions, and verify that your applications work correctly. If you miss this step, your staged or live application might stop working after the upgrade. For more, see *Visual Builder apps on obsolete VB runtime versions in Known Issues for Oracle Integration 3*.
- If you use the "delegate authentication" authentication type in any service connections or backends, you must switch the authentication type to "Oracle Cloud Account". The "delegate authentication" authentication type is deprecated. After you change the authentication type, test your service connections and apps to confirm that they still work correctly.
- If you use Visual Builder with your own Oracle database instance (BYODB), there are several steps you need to take to prepare for upgrade:
 1. Keep Autonomous Transaction Processing (ATP) up and running during the upgrade.
 2. If you have an allowlist (also known as an ACL) configured in ATP, add the Visual Builder VCN OCID to the ATP allowlist.
 3. Ensure that the user name and password for your ATP instance are configured correctly in Visual Builder.
 4. Reset the expired password or wallet that is used for connecting to ATP.
 - a. Download the latest wallet. See [Download Database Connection Information in Using Oracle Autonomous Database Serverless](#).
 - b. Upload the wallet on the instance's **Tenant Settings** page. See [Update Your ATP Wallet and Reset an Expired Password](#).

Tasks to Complete After the Upgrade

- Configure an IAM policy for Visual Builder in Oracle Integration 3. See [Set the IAM Policy for Managing the Visual Builder Instance](#).
- Update the allowlist and network access rules to add the Visual Builder service VCN OCID in Oracle Integration 3. See [Allow Your Instance to Access Services](#).
- Update backends and service connections.
 - Oracle Integration 3 does not accept Basic Auth, and recommends OAuth. For more details, see [When is Basic Authentication Supported in Oracle Integration 3?](#)
 After migration, you need to update backends and service connections to the Oracle Integration REST API that use Basic Auth, to change them to use an appropriate OAuth mechanism (OAuth 2.0 Resource Owner Password is the most similar to Basic Auth). For backends and service connections connecting to Oracle Integration REST API using "Oracle Cloud Account" authentication, you don't need to change the authentication.
 - If your backend (or service connection) had the connection type set to "Dynamic, Service supports CORS", confirm this is set to "Always Use Proxy". This is to avoid any Chrome 119 issues that might arise due to redirects of the Oracle Integration REST API.

- If your service connections to Integration REST APIs were created from the catalog, they will continue to work after the migration (provided Basic Auth is replaced with a suitable OAuth mechanism, and the connection type is updated to "Always Use Proxy").
- Service connections to the Oracle Integration design-time (or factory) APIs, created using the endpoint flow in Visual Builder, that are not listed in the catalog, will not work after the migration, and you need to update them.

An example of such an endpoint is a lookup endpoint (see [Lookups REST Endpoints](#) in *REST API for Oracle Integration 3*). When you test them in the service connection's Test tab, these endpoints will typically give error code HTTP 403, with the error "Unable to verify URL against allowed list". See [SendRequest](#) in *REST API for Oracle Integration 3* for more on the design-time URL and factory APIs.

To rework these service connections using a custom backend:

1. Create a custom backend called `icsFactoryApi` (for example). For the URL, use the design-time URL (for example, `https://design.integration.region.ocp.oraclecloud.com`), and use OAuth authentication (generally OAuth 2.0 Resource Owner Password). Set the connection type to "Always Use Proxy" (recommended).
 2. For existing service connections based on design-time APIs:
 - a. Change the URL from `vb-catalog://backends/ics` to point to `vb-catalog://backends/icsFactoryAPI`. (If the URL is `https://<integration_base_url>`, then replace `<integration_base_url>` with `vb-catalog://backends/icsFactoryAPI`).
 - b. Navigate to the Request tab, and then add the static query parameter of "integrationInstance", and add the appropriate value.
 3. The static query parameter introduced this way becomes part of the visual application code. This parameter will be different for different Integration instances, so you would need to update the parameter if you import the app to another instance (for example, a prod instance). If you use a build pipeline, you would need to modify the parameter in the packaging jobs.
- (Optional, but recommended) Set and store the details about the authentication type, connection type, and credentials on the backend, and have all the service connections based on the backend. Defining details only on the backend can help you manage the details, so you don't have to duplicate them as much. See the blog post [Streamlining Service Connections to use Backends](#) for more information.
- PWA apps installed on user devices will stop receiving automatic updates after your Visual Builder instance is upgraded from Oracle Integration Gen 2 to Oracle Integration 3.

After the upgrade:

1. Republish the PWAs.
2. Provide users with the new URL. You should instruct the users that they need to uninstall the old apps and install the new ones from the new Visual Builder location in Oracle Integration 3.

The apps will receive automatic updates after they are re-installed. For more, see PWA app no longer receives updates automatically after migration to OIC 3 in *Known Issues for Oracle Integration 3*.

- If your application code has any references to the Oracle Integration Gen 2 Visual Builder URL, you need to update them to the new Oracle Integration 3 VB URL after the upgrade for your apps to work correctly. For more, see [Unable to access VB business object URLs](#)

from service connections in live app and Custom code accessing pre-upgrade Visual Builder URL failing in *Known Issues for Oracle Integration 3*.

4

Configure Tenant Settings

After a Visual Builder service instance is created, an identity domain administrator assigns one or more users the Visual Builder Administrator role for the service instance. A Visual Builder Administrator can manage and set general options for applications in the service instance.

Manage Applications in the Service Instance

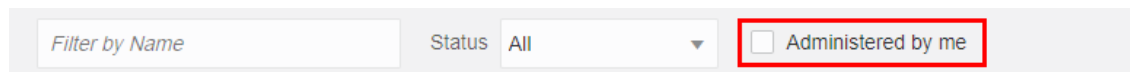
The Visual Builder Home page helps you manage visual applications created on the current Visual Builder service instance, as well as apps deployed to the instance from a different VB instance.

Each row on the Home page represents an application created on the Visual Builder instance, or deployed to the instance:

| Name | Status | Origin | Version | Recent activity |
|---------------------|-------------|----------------|----------|-----------------|
| Sample JET | Stage | Visual Builder | 1.0.1 +1 | 12/15/2023 |
| CallingVBInt_KB-1.0 | Stage | Visual Builder | 1.0 | 3/24/2023 |
| vp_new | Development | Visual Builder | 1.0 | 2/15/2023 |
| TestRecipe_KB | Live | Visual Builder | 1.0 | 2/13/2023 |
| TESTMVRP_KB | Live | Visual Builder | 1.0 | 2/13/2023 |

The first row in the image above represents an application created on this VB instance, which can be managed using the in the right column. When a row is selected as in the image above, you can click **Move to Trash** to delete the application from the Identity Domain. You have 30 days to recover the application after deleting it. For more, see *Delete a Visual Application* in *Developing Applications with Oracle Visual Builder in Oracle Integration Generation 3*.

If you're an administrator, you can manage (which includes deleting) any of the apps shown on the Home page. To see the full list of applications, select the **Administered by me** checkbox next to the Status dropdown list.



If you're a developer, you can manage the apps that you've created in this instance of Visual Builder, or if you're a team member of the app. You will not see the **Administered by me** checkbox unless you have the role of administrator.

IDCS Client Applications

Each time a visual application is created on a Visual Builder instance provisioned with OIC, a companion *client application* is automatically created in IDCS. When you stage or publish the app to a different server governed by the same IDCS instance, another client application is created. This means there may be several client applications for the same visual application running on IDCS for the lifespan of the app.

If you have IDCS administrator privileges, you can see these client apps on the IDCS console, but you should not need to *manage* them in any way. When a visual application is removed from the server, the associated client apps are removed from IDCS. You may need to *interact* with the client apps if you want to set login-related policies for an instance—for example, to enable the **Keep me signed in** policy, or to assign the users and groups who can access the application.

When you want to view client applications in the IDCS console, make sure that you are looking at the correct IDCS app for the instance. VB instances provisioned with OIC share the OIC instance's IDCS app, and the client apps are listed in that OIC instance's IDCS app. You can look at the URL of the service host name to help determine if a VB instance was provisioned with Oracle Integration 3 or Oracle Integration Generation 2:

- When the URL of the service host name contains `<SUB-DOMAIN>.builder.ocp.oraclecloud.com`, the VB instance was provisioned with Oracle Integration 3 (or it's an Oracle Visual Builder instance).
- When the URL contains `<SUB-DOMAIN>.integration.ocp.oraclecloud.com`, the VB instance was provisioned with Oracle Integration Generation 2.

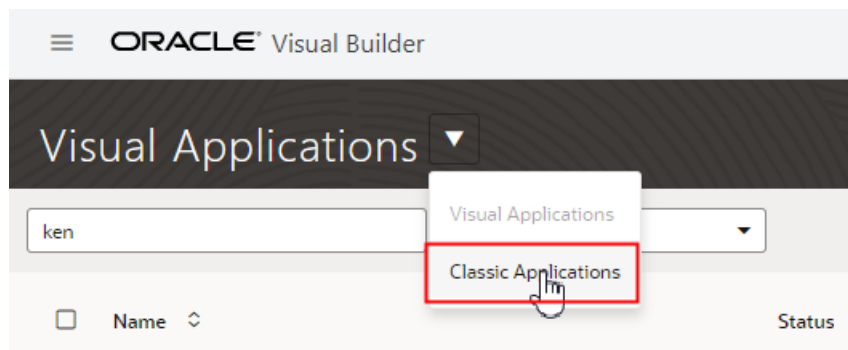
 **Note:**

For Oracle Visual Builder instances not provisioned with Oracle Integration, the URL also contains `<SUB-DOMAIN>.builder.ocp.oraclecloud.com`, however, the instances will have their own IDCS app, and the client apps are listed in the Oracle Visual Builder instances' IDCS apps.

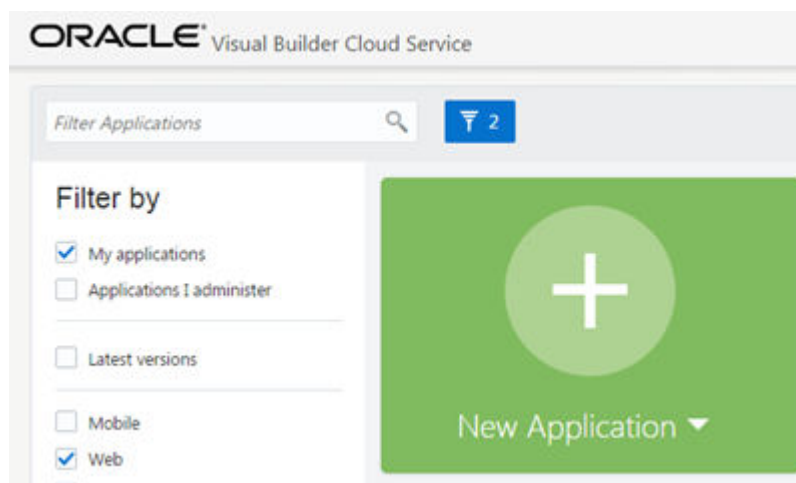
For more, see *Typical Workflow for Managing Oracle Identity Cloud Service Applications in Administering Oracle Identity Cloud Service*.

Visual Builder Classic Applications

If you have any classic applications (apps that have the older Visual Builder project structure), open the Visual Applications dropdown list in the header and select **Classic Applications**.






On the Home page for classic applications, administrators can select the **Applications I administer** checkbox in the Filter by pane to display the applications where they are not a team member.




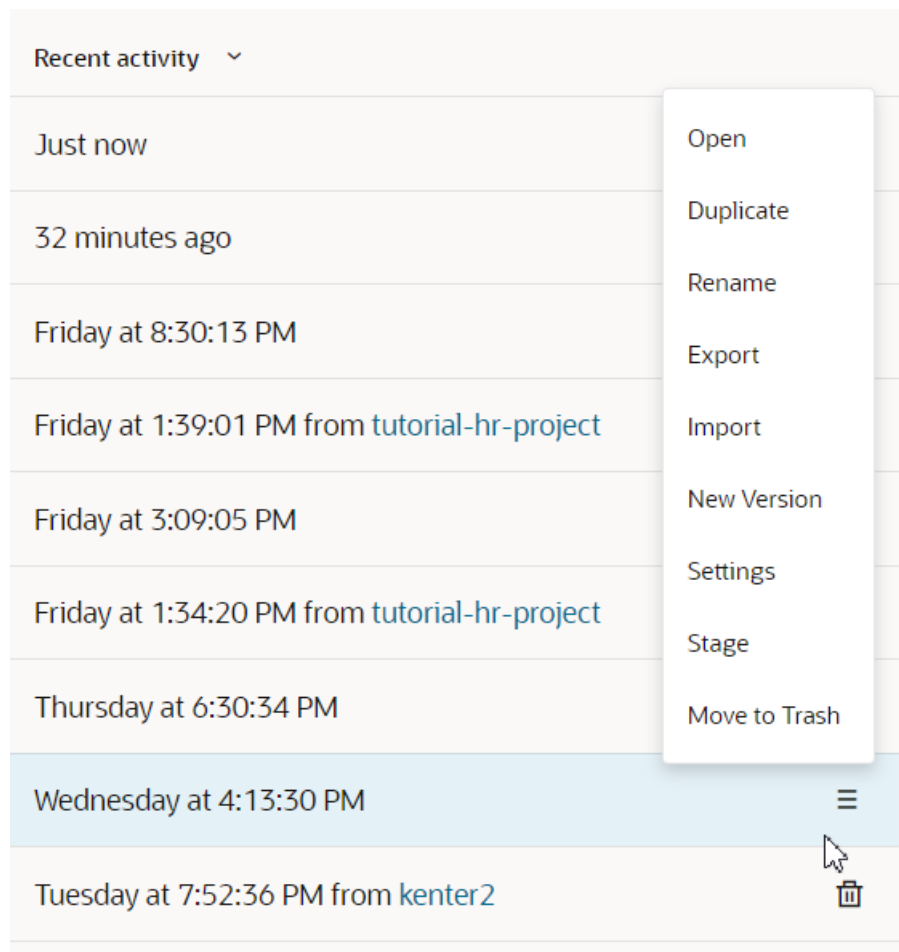
Manage Applications Created on the Instance

The table of applications on the Home page includes visual applications created on the Visual Builder instance. Each of these applications has an Options menu in the right column that developers and administrators can use to manage it, for example, to add and remove team members, and to open, stage and publish the application.

You can tell which applications were created on the instance by looking for  in the right column. If you see a  button instead of , the application was created on a different instance of Visual Builder. If an application was created on a different instance, you'll need to open the application on the instance where it was created and manage it there.

To manage an application created on the current Visual Builder instance:

- Click  in the right column of the application, and then select a task in the Options menu:



The Options menu displays commands for managing the application (based on the application's status, some commands might be hidden):

| Menu Item | Description |
|-------------|--|
| Open | Opens the development version of the application |
| Duplicate | Creates a clone of this version of the application, including the content of the database. |
| Rename | Opens a dialog box where you can change the name of the application. |
| Export | Creates a ZIP archive of the application that can be imported as a new application. When exporting the application, you can choose if you want the exported archive to include the data stored in your business objects. |
| Import | Opens a dialog that you can use to create an application by uploading an application archive (ZIP or OVB) from your local system. |
| New Version | Creates a new version of the same application. By default the new version is a development version. Version numbers are automatically increased incrementally. |
| Settings | Opens an editor for configuring the application's settings and viewing the application API URLs. Each application version has a dedicated Settings editor. |


| Menu Item | Description |
|---------------|---|
| Stage | Opens a dialog box where you can specify the database option for the staged application. When an application is staged, a link to the staged version is displayed in the tile. |
| Publish | Opens a dialog box where you can specify the database option and publish the staged version of your application. |
| Lock / Unlock | Enables you to lock a live application to prevent any users from using the application. You would usually use this command when you are going to update the live application with a newer version. The Unlock option is displayed only when the live application is locked. |
| Rollback | Rolls back the live version to the previous live version. This is only available for the current live version, and there must be an older live version of the app. |
| Move to trash | Deletes the application from the Identity Domain. You have 30 days to recover the application after deleting it. For more, see Delete a Visual Application in <i>Developing Applications with Oracle Visual Builder in Oracle Integration Generation 2</i> . |

Access Tenant Settings

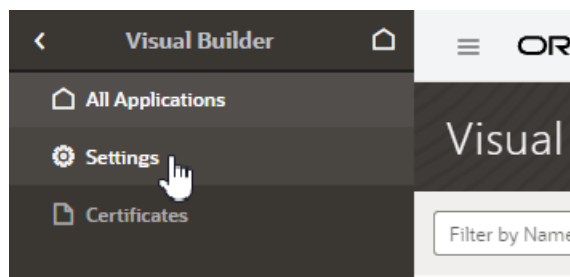
An instance administrator can access the Tenant Settings page for managing the instance's global settings from any Visual Builder page.

The Tenant Settings page contains three tabs: General, Tenant Database and Services. The General tab has panels for configuring security settings, specifying Access Denied messages, and configuring the Component Exchange details. You use the Tenant Database tab to switch to an Oracle database and to see how much database space your applications are using. You use the Services tab to add and edit the backend services that are accessible to apps in the tenant.

To open an instance's Tenant Settings page:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** .
2. Click **Settings** in the main menu.

If you are developing visual applications, open the main navigation pane on the Home page and select **Settings**.



The settings available for the instance are grouped in the General, Tenant Database and Services tabs on the page.

The screenshot shows the 'Tenant Settings' interface with a dark header and a light body. The 'General' tab is selected. The 'Security' section includes two checkboxes: 'Allow only secure applications to be created' (unchecked) and 'Only Visual Builder users can access secure applications' (unchecked). Below these are two text input fields: 'No Visual Builder Access' (containing 'Enter redirect URL') and 'Access Denied Message'. The 'Component Exchange' section has three text input fields: 'Server URL' (containing 'https://exchange.oraclecorp.com/api/0.2.0'), 'Username' (containing 'Enter Component Exchange Service Username'), and 'Password' (containing 'Enter Component Exchange Service Password'). At the bottom, there is an 'Allowed Origins' section with a '+ New Origin' button and a text input field for 'Origin Address'.

Choose Your Instance's Update Window

Functional updates for Visual Builder are provided in two windows, which are typically two weeks apart. You can select when you want an instance updated by selecting either `Window 1` or `Window 2`. We recommend that non-production instances be updated in the first window (`Window 1`) and production instances in the second window (`Window 2`). This allows you to test your applications in your test and development environments before the update is applied to your production environment.



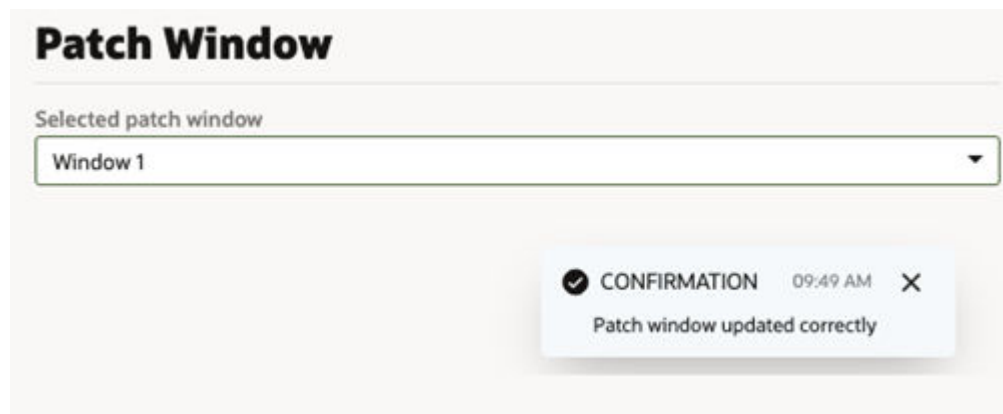
Note:

Oracle automatically sends notifications to the instance's account administrator each time it will be updated, confirming the instance's next update window. Once we send out the notification, it's too late to change your window for that update. If you do make a change, it won't be applied until the following update.

To set the update window option:

1. Open the Visual Builder instance's **Tenant Settings** editor.
2. In the General tab, select the window option in the Patch Window dropdown list.

There are only two options: `Window 1` and `Window 2`. The default update window is `Window 1`.



Configure Security Options for Applications

Administrators can use the Security panel in the Tenant Settings page to require authentication for all applications in the instance.


When an administrator enables the **Allow only secure applications to be created** option, all published and staged applications in the instance will require user authentication. When the option is enabled, users must log in to access the applications in the instance, even if anonymous access is allowed in the application's settings. When the option is not enabled, applications can be created that allow access to anonymous users.

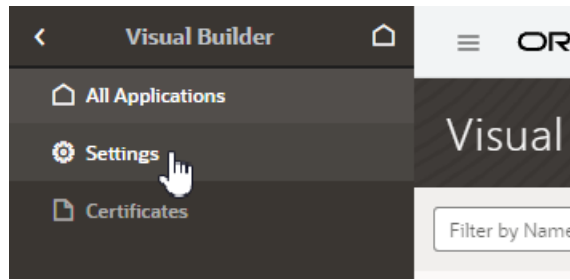
When an application has the default security settings, any user with a valid login can access the pages in an application. A developer can modify the default security settings to define the roles that can access applications, pages and components.

When the secure application option is enabled, an administrator can enable the **Only Visual Builder Users can access secure applications** option so that only Visual Builder users (those assigned the default Service User role) can access the staged and published applications in the instance. For example, this allows you to configure security so that users assigned the Visual Builder Developer role can access the designer, but can't access the published application and data because they are not assigned the Visual Builder Service User role.

An administrator can also use IDCS roles when configuring the instance's security so that a user's access is limited to just the secure applications. Users assigned the selected IDCS role would be able to access the applications, but would be prevented from accessing Visual Builder or Oracle Integration resources external to the application, such as other Oracle Integration integrations.

To configure the security options for all applications in the instance:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** .
2. Click **Settings** in the navigation menu to open Tenant Settings.



3. In the **Security** panel, enable **Allow only secure applications to be created**.

Anonymous users can't access the applications when this secure applications option is enabled.

4. Select the **Only Visual Builder Users can access secure applications** option if you want to allow only Visual Builder users (users assigned the Service User role) access to the applications.

To change the users allowed to access the application to those assigned a specific IDCS role *instead of* those assigned the default Service User role, select the IDCS role in the dropdown list under **Allow access to application to users in role**. This option is only available when both of the other security options are enabled.

5. Specify what users denied access to the secure application will see:
 - a. Enter the URL you want the users redirected to when they can't access the app.
 - b. Enter an Access Denied message that they will see when denied access to a page in the app.

Assign Roles for Users to Access an Application

Administrators must assign roles to users, so they have the permissions required to access Visual Builder applications.

Privileges associated with a user role determine what tasks users assigned those roles can perform. See *What Users Can Do in Visual Builder by Role* and *Manage Access and Assign Roles in Provisioning and Administering Oracle Integration 3*.

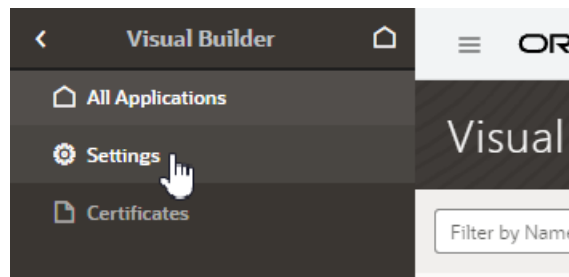
Set Page Messages for Access Denied Errors

Administrators can use the instance's settings page to specify a URL that users are navigated to when they are denied access to an application or page.

Authenticated users might see an Access Denied page or message when they attempt to access an application or page in an application that their user role is not permitted to access. Administrators can set the default page or message that users see when they are denied access to an application or page. Access Denied messages that are set at the application level in the General Settings of an application will override messages set in the instance's settings page. The default Access Denied page and message is used if the message options in this panel are not set.

To specify an Access Denied page or message for applications in the instance:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.



3. In the **Security** panel, type a URL that users are directed to when denied access to an application.

The URL that you specify is used as the Access Denied page for all applications in the instance and should be accessible to users who are not logged in.

No Visual Builder Access

Enter redirect URL

"Access Denied" Message

 **Note:**

If you are configuring settings for classic applications, the Access Denied settings are set in the **Messages** panel.


4. Type the message that you want users to see when they are denied access to a page.
The message that you enter will be displayed in the Access Denied page for all applications in the instance except for those where a message was set at the application level in the application's General Settings page.

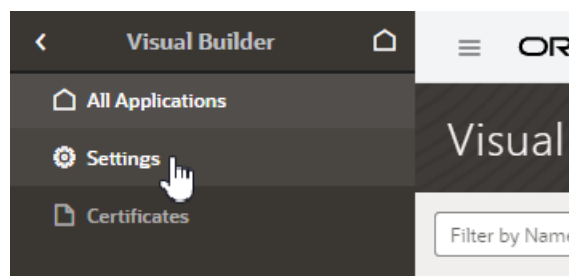
Allow Other Domains Access to Services

Use the Global Settings page to specify the domains that are permitted to interact with services in your instance.

Cross-Origin Resource Sharing (CORS) is a mechanism that enables you to specify the domains that are allowed to exchange data with applications in your instance. By default, incoming requests from domains not on your instance's list of allowed origins are blocked from accessing application resources.

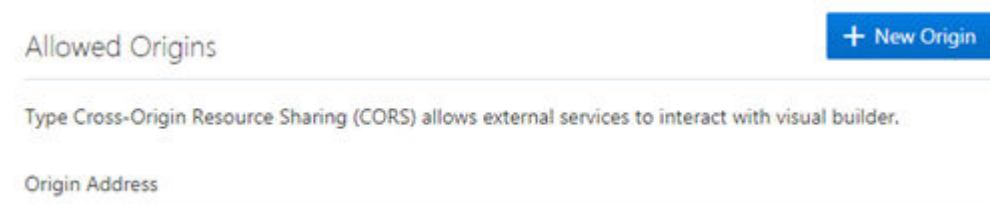
To add a domain to the list of allowed origins:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** .
2. Click **Settings** in the navigation menu to open Tenant Settings.



3. In the **Allowed Origins** panel, click **New Origin** and type the URL of the domain that you want to allow. Click **Submit**.

The URL must be a fully-qualified domain, meaning it must contain `http://` or `https://`, for example, `https://myoracle.cloud.service`. You must explicitly enter each fully-qualified domain that you want to allow. To allow both `http://` and `https://` connections from a domain, you would need to add both domains (`https://myoracle.cloud.service` and `http://myoracle.cloud.service`).



The Allowed Origins panel lists all origins that are permitted to retrieve information from the instance.

Allow Your Instance to Access Services

If your Visual Builder instance needs to access an external service, your instance needs to be included in the service's allowlist (formerly a whitelist).

A service typically uses an Access Control List (ACL), called an allowlist, to restrict the networks and services that are allowed to access it. Only users from an IP address or Virtual Cloud Network (VCN) on the allowlist are allowed access to the service. The allowlist restrictions are in addition to the standard authorization mechanisms, such as user credentials, which are always in place.

Any Visual Builder instance that requires access to an external service, such as a REST web service, must be on the external service's allowlist. To get on a web service's allowlist, you'll need to work with the web service's administrator to add an ACL access rule for your VB instance. This may require filing a Service Request with the web service's administrator. You'll typically only need to do this when creating a new VB instance that will require access to a service, or when you plan to start using a new service in a VB instance. A VB instance can be added to an allowlist at any time, even before the instance has been created.

Depending on the location and type of the service your VB instance needs to access, you'll need to provide the service's administrator with:

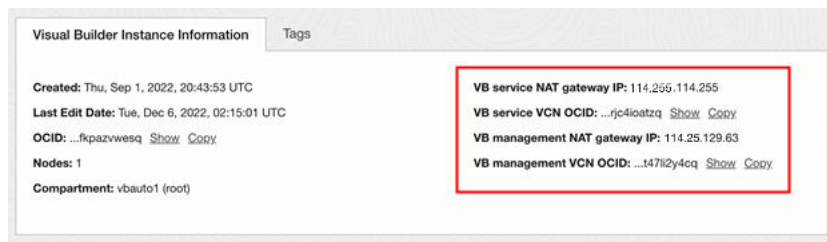
- the Visual Builder *service* VCN,
- the Oracle Cloud ID (OCID) of the Visual Builder service VCN, or
- the NAT gateway IP address of the Visual Builder service VCN.

A VB instance's service VCN, OCID and NAT gateway IP address are determined by the instance's *region*. For example, `iad-vb-isovcn` is the VB service VCN for instances in the Ashburn region. For details on what these are, see [Overview of VCNs and Subnets](#) and [NAT Gateway](#) in the *OCI Documentation*.

 **Note:**

Visual Builder instances that use an Oracle DB service (ATP, DBaaS) will **also** have a VB *management* VCN. The VB management VCN OCID or NAT IP **must also** be added to the service's allowlist. Access from the VB management VCN is required so that schemas related to the VB service can be updated, for example, when patches or updates are applied to the instance.

You can view an instance's VB service NAT gateway IP and VCN OCID in the instance's Visual Builder Instance Information tab in the OCI console. If the instance also has a VB management NAT gateway IP and VCN OCID, they will also be displayed in the tab:



The instance details you need to provide in the Service Request will depend upon the location and type of the service your instance needs to access. In the following table you can see the VB instance details you'll need to provide to allow VB to access the different types of external services.

| To access... | You'll need... |
|---|--|
| A REST web service located in Oracle Service Network (OSN) (such as ORDS) | <p>You'll need to provide:</p> <ul style="list-style-type: none"> the VB <i>service</i> VCN OCID. <p>The service administrator needs to configure one access rule, to allow access from the VB runtime service VCN.</p> |
| An autonomous database located in OSN, like ATP | <p>You'll need to provide:</p> <ul style="list-style-type: none"> the VB <i>service</i> VCN OCID, and the VB <i>management</i> VCN OCID <p>The service administrator needs to configure two access rules, to allow access from the VB runtime service VCN and the VB management VCN.</p> |
| an external REST web service | <p>You'll need to provide:</p> <ul style="list-style-type: none"> the NAT gateway IP address for the VB <i>service</i> VCN <p>The service administrator needs to configure one access rule, to allow access from the IP address of the NAT gateway of the VB runtime service.</p> <p>An access rule configured for the NAT gateway is used when the service is not in the same region and OSN as your instance.</p> |

| To access... | You'll need... |
|----------------------------|---|
| an external DBaaS database | <p>You'll need to provide:</p> <ul style="list-style-type: none"> the NAT gateway IP address for the VB <i>service</i> VCN the NAT gateway IP address for the VB <i>management</i> VCN <p>The service administrator needs to configure two access rules, to allow access from the VB runtime service VCN NAT gateway and the VB management VCN NAT gateway.</p> <p>Access rules configured for the NAT gateways are used when the service is not in the same region and OSN as your instance.</p> |

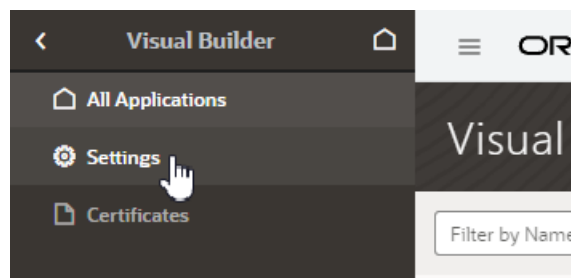
Inspect Database Usage

An administrator can view how much space in the tenant's database is being consumed by each of the tenant's applications.

The capacity of the tenant's database is 5GB, so by viewing the database usage you can see how much of the database's capacity remains. If you require more than 5GB of storage, you can [Switch to Your Own Oracle DB Instance](#).

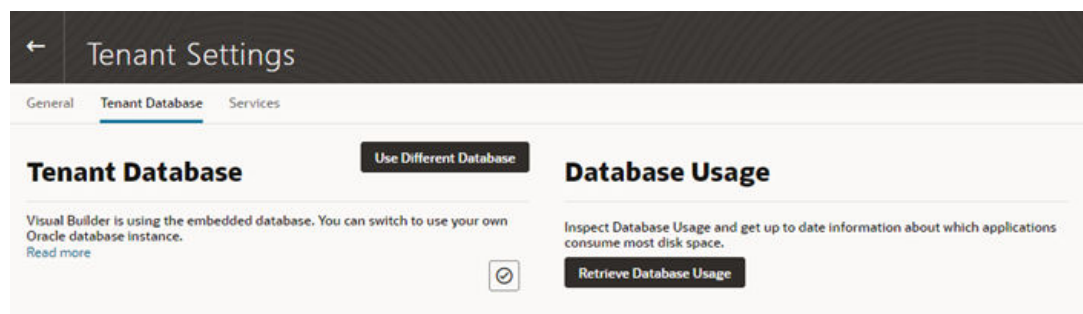
To inspect your instance's database usage:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.

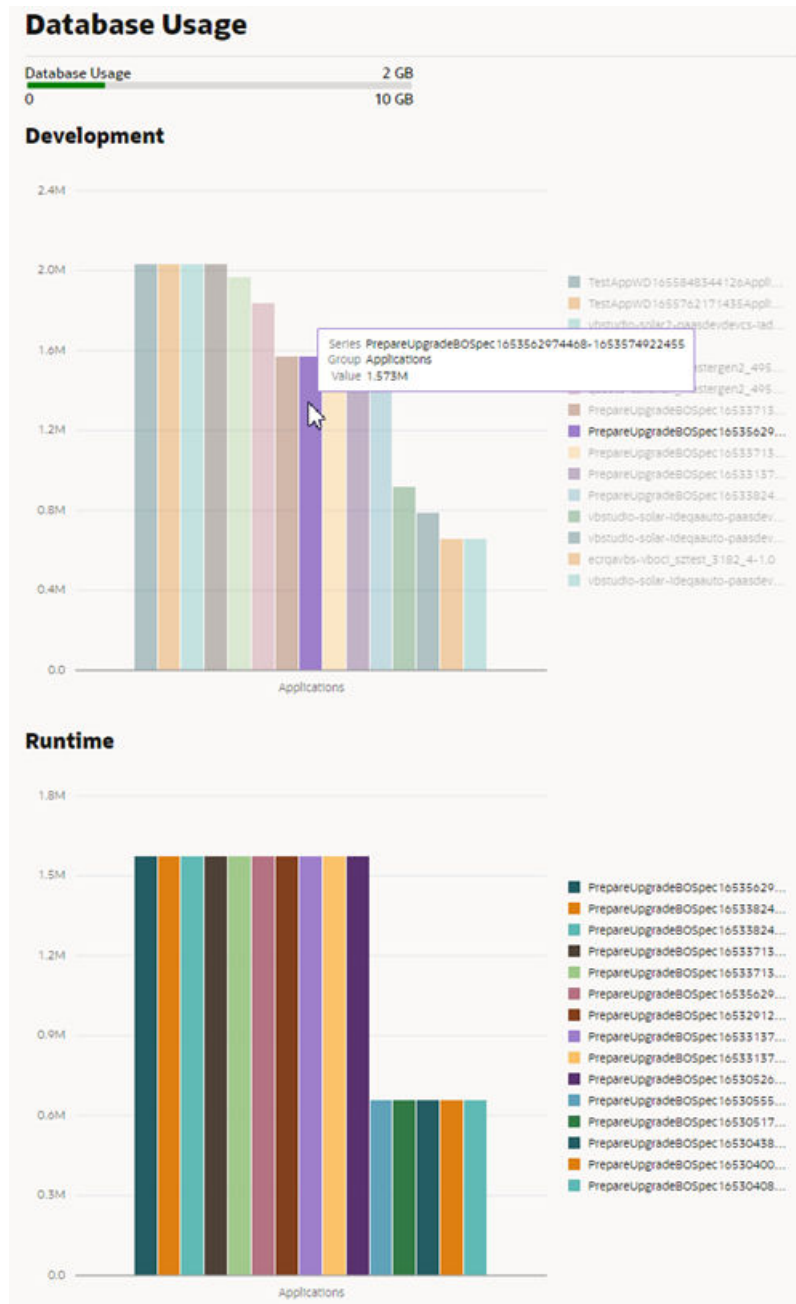


3. Open the Tenant Database tab.

The Tenant Database has two panels: Tenant Database and Database Usage.



4. Click **Retrieve Database Usage** in the Database Usage pane.



The Database Usage meter in the panel shows how much of the database's capacity is currently used. The data usage is rendered in two charts:

- The Development chart shows the space used for storing the business objects in each application in the design-time.
- The Runtime chart shows the space used for each of the staged and published apps.

Switch to Your Own Oracle DB Instance

The database provisioned with your Visual Builder instance is used to store data for your business objects and your app's metadata, but this database has a 5GB limit and you can't access the data in the objects using regular SQL.

If the 5GB limit is insufficient for your tenant schema, you can configure your instance to use an Oracle DB instance that has more space instead of the default database. If you choose to switch to an Oracle DB instance, the database must be publicly accessible. You can connect to an Oracle DBaaS or Autonomous Transaction Processing (ATP) database instance. Using an ATP database will give you more space and direct SQL access to the objects VB creates. You can also use a [Free Forever Oracle ATP](#), which provides 20GB of storage for free.

To use a different Oracle DB instance, you use a wizard in the Tenant Settings to create a connection to the database instance and export the applications stored in the tenant's current database.

If you decide to use JDBC to connect to your DBaaS instance, you must include the privileges required to enable the ADMIN user to create a tenant schema. The following SQL shows the grants that are needed:

```
CREATE USER [adminuser] IDENTIFIED BY [password];
GRANT CONNECT, RESOURCE, DBA TO [adminuser];

GRANT SELECT ON SYS.DBA_PROFILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_USERS TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_DATA_FILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_SEGMENTS TO [adminuser] WITH GRANT OPTION;
```

If you decide to use ATP, you'll need to include the `wallet.zip` file in the wizard in addition to the connection info. You might want to create a new ATP ADMIN user with the correct admin privileges. The following SQL statement shows how to create a second ATP ADMIN user in SQL*Plus or SQL Developer.

```
DROP USER [adminuser] CASCADE;
CREATE USER [adminuser] IDENTIFIED BY [password];
GRANT CREATE USER, ALTER USER, DROP USER, CREATE PROFILE TO [adminuser] WITH
ADMIN OPTION;
GRANT CONNECT TO [adminuser] WITH ADMIN OPTION;
GRANT RESOURCE TO [adminuser] WITH ADMIN OPTION;
GRANT CREATE SEQUENCE, CREATE OPERATOR, CREATE SESSION,ALTER SESSION, CREATE
PROCEDURE, CREATE VIEW, CREATE JOB,CREATE DIMENSION,CREATE INDEXTYPE,CREATE
TYPE,CREATE TRIGGER,CREATE TABLE,CREATE PROFILE TO [adminuser] WITH ADMIN
OPTION;
GRANT UNLIMITED TABLESPACE TO [adminuser] WITH ADMIN OPTION;
GRANT SELECT ON SYS.DBA_PROFILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_USERS TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_DATA_FILES TO [adminuser] WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_SEGMENTS TO [adminuser] WITH GRANT OPTION;
```

 **Note:**

If you get an error `Failed to verify the target database` in the Change Tenant Database dialog when switching the database, it might be because the database is not reachable (Visual Builder cannot reach databases in private subnets), or because you don't have the required privileges.

If you see the error, confirm that the ADMIN user (`adminuser`) has the required privileges. You might also need to assign the `SYSOPER` and `SYSDBA` roles to the ADMIN user:

```
GRANT SYSOPER, SYSDBA TO [adminuser];
```

You can run the following query to confirm the ADMIN user has the necessary privileges:

```
select * from v$pwfile_users;
```

In the wizard you need to select and export all the applications in your instance that you want to keep. After confirming that your instance is using the new database instance, you must import the exported applications into Visual Builder to save them in the new database instance.

 **Note:**

If you have live applications already on the instance:

- Before switching to a new database, make sure to backup the data in their business objects using the export options in the Visual Builder data manager. You'll then be able to import that data back into the new apps you'll create from the application archives you export in the wizard.
- Lock the live applications before changing the settings of your instance's database to prevent users from using them during the migration process. You can unlock the applications when the migration process is finished. You lock and unlock live applications in the Application Options menu on the Visual Builder Home page. See *Manage an Application* in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

To switch to a different Oracle DB instance:

1. Open the Tenant Database tab.

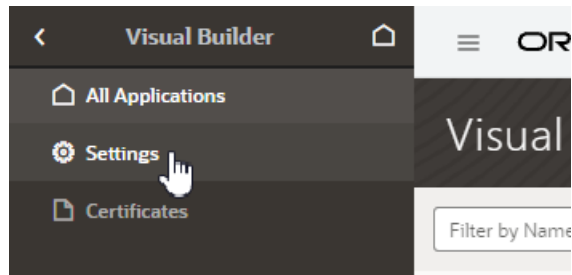
You can open your instance's Tenant Database tab from the instance Home Page, or by entering the URL directly in the browser window. It might be quicker to enter the URL directly if there is a problem loading the Home Page, for example, if the wallet is expired.

- To open the Tenant Database tab using a URL, type the following in the browser's URL field:

```
https://<instance-url>/ic/builder?root=settings&settingsSection=tenant-database
```

In the URL above, replace *<instance-url>* with your instance's URL.

- To open the Tenant Database tab from the Home Page:
 - a. On the Visual Builder Home Page, click **Navigation Menu** ☰ in the upper-left corner of the Visual Builder title bar.
 - b. Click **Settings** in the navigation menu to open Tenant Settings.



- c. Open the Tenant Database tab.
 2. In the Tenant Database tab, click **Use Different Database** in the Tenant Database panel to open the Change Tenant Database wizard.

In the Change Tenant Database wizard you supply the details for the connection to your Oracle DB instance.

Change Tenant Database

Cancel 1 Define Database 2 Export Applications Next >

Connection Type
Oracle Autonomous Transaction Processing Cloud Wallet

Upload Wallet
Upload a zip file or drag one here

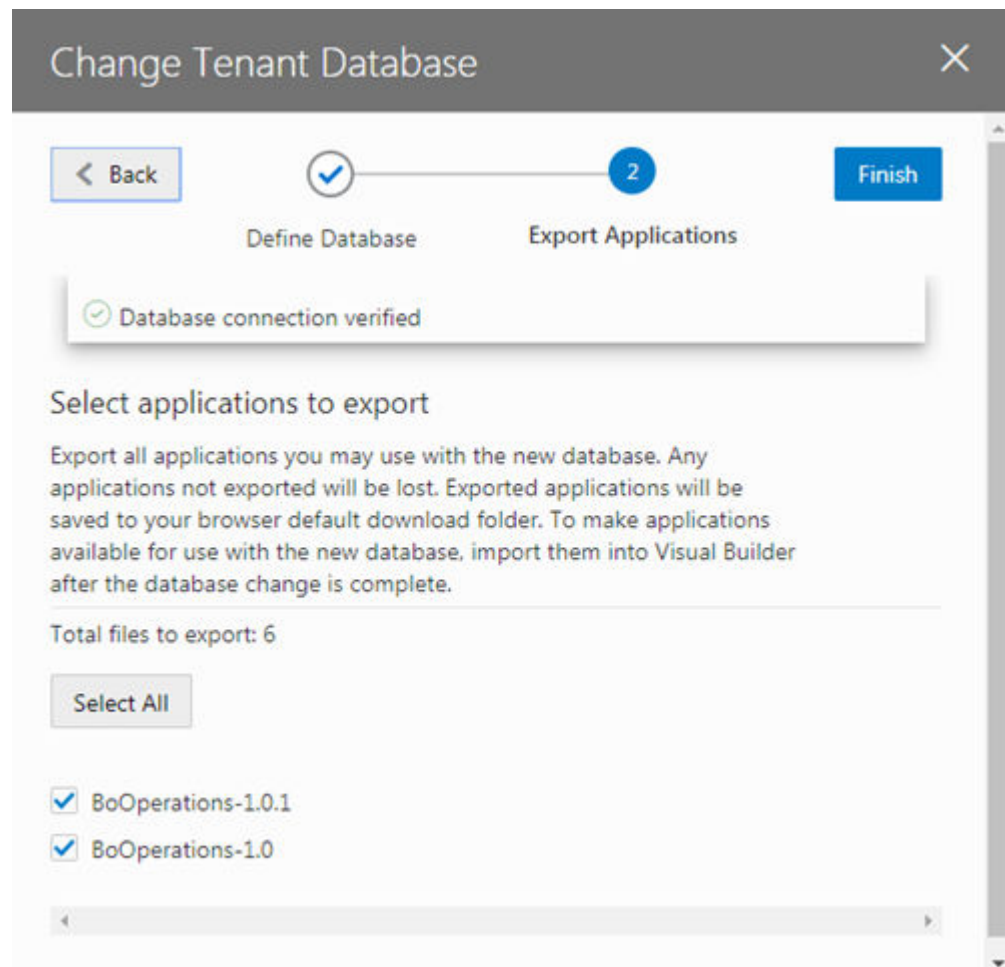
Wallet Password

TNS Name

DBA User Name Please make sure the user has DBA privilege

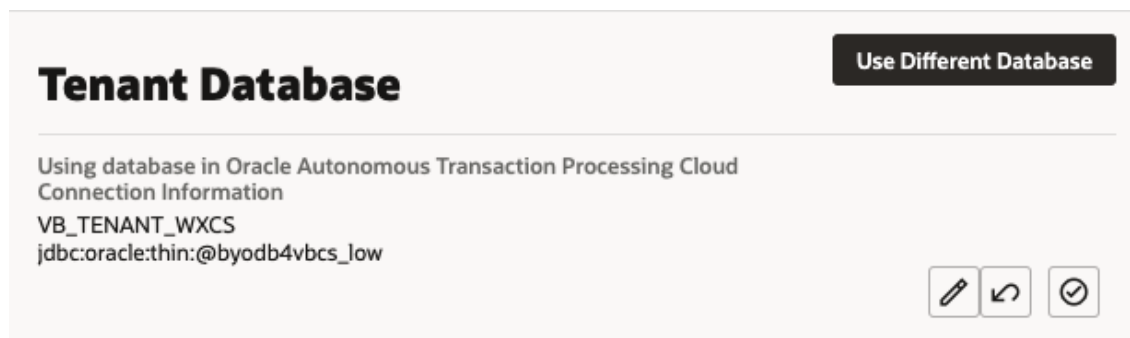
Password The user credential will be used to create schema only, and will not be saved

3. Select a Connection Type in the drop-down list.
You can connect to your Oracle DB instance using either JDBC or an ATP Cloud Wallet.
4. Provide the details for connecting to your database. Click **Next**.
The details you need to provide will depend upon the type of connection you selected.
5. Select all the applications that you want to export. Click **Finish**.
You must select and export all the applications that you want to keep. Any applications that are not exported will be lost.




When you click Finish, the applications that you selected are downloaded to your local file system. Exported application archives include the details about the application's user roles, and they will be available when you re-import your app into the new database.

After switching the database, the Tenant Database pane displays the connection information for your tenant's database. In the following image you can see that the instance is now using an Autonomous Transaction Processing (ATP) database instance.



 **Note:**

If you decide to revert back to using the embedded database, you can click  in the Tenant Database pane. You'll be prompted to confirm that you want to switch to using the instance's embedded database instead of the current one.

When you revert to using the embedded database, the visual applications in your current database are not transferred automatically. You need to export the apps you want to keep before switching the database, and then import them into the embedded database.

Visual Builder automatically manages the schemas and tables it uses for apps and business objects in your new DB, so you don't need to do anything further.


If you would like to access the business objects using SQL, you'll find that VB creates users/schemas with names that start with `VB_` followed by randomly generated strings. By examining the data dictionary you'll be able to find the users that represent specific apps. Note that you'll see separate schemas for dev, stage, and published instances of an app. The schemas for the dev and test instances will be re-created with different names with every new version of the app that you create. If you want to prevent the schema name for a published app from changing, when you publish new versions of the app you should choose the option to not replace the data.

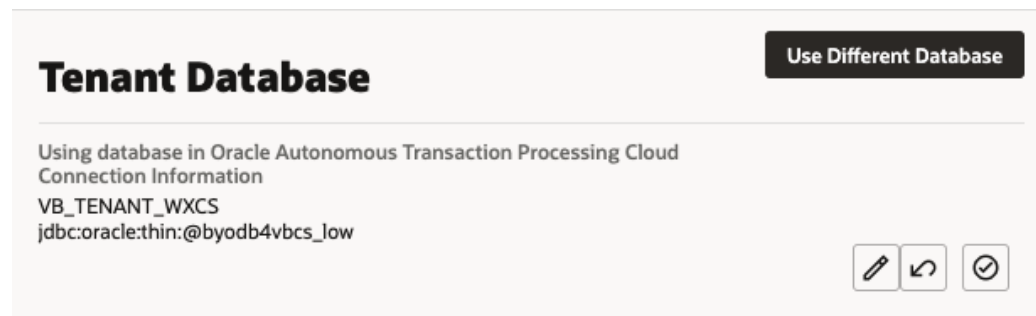
 **Note:**

Instead of having Visual Builder create and manage schemas, you can make a schema that already exists in your database available to applications, so developers can create business objects based on existing DB tables and views. In this case, only one schema is used for the app's dev, staged, and published instances. See [Make Schemas in an Oracle DB Instance Available to Applications](#).

Switch From One ATP Database to Another

It's not possible to switch from one ATP database to another directly, so you'll first need to switch from your ATP database back to the embedded database. You can then use the Change Tenant Database wizard to switch from the embedded database to the new ATP database.

1. In Visual Builder, export each of your visual applications and save them to your local system.
For details, see [Export a Visual Application](#).
2. Revert to the embedded database.
 - a. Open the instance's Tenant Settings page, and then open the Tenant Database tab.
 - b. Click  in the Tenant Database pane to revert to the embedded database.



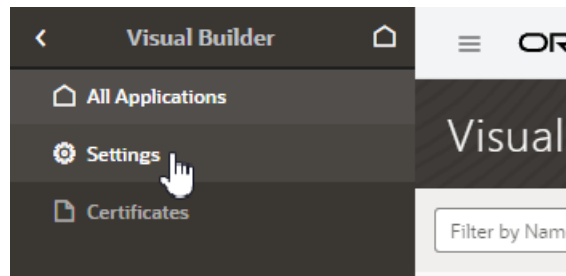
- c. When prompted, confirm that you want to switch to the embedded database.
3. Switch to the new ATP database.
Follow the steps in [Switch to Your Own Oracle DB Instance](#) above to switch from the embedded database to your new ATP database.
4. Import the applications you saved to your local system into the new ATP database.

Make Schemas in an Oracle DB Instance Available to Applications

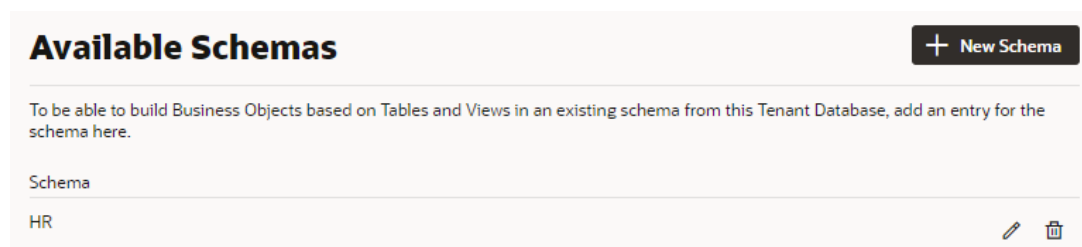
When you connect an Oracle database instance with your Visual Builder instance, application developers can use schemas predefined in the tenant database to create business objects based on existing tables and views for an application. But for developers to access these schemas, you'll first need to make them available to applications.

To make a tenant database's existing schema available to applications:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** ☰.
2. Click **Settings** in the navigation menu to open Tenant Settings.



3. Click **+ New Schema** in the Available Schemas panel.



4. Enter a name for the schema and click the check mark icon.

After the schema is added, you can edit its name or delete it entirely, but remember any changes you make might break applications that use the schema.

Schema that exists in the tenant database and has been added to the list of available schemas will become available for selection in an application's Settings editor (under Schema Selection in the Business Objects tab).

Update Your ATP Wallet and Reset an Expired Password

If you switch to your own Oracle DB instance and the credentials you use to access the instance expire, you can use the Update Tenant Database Connection dialog box to update your ATP wallet and renew expired credentials.

To regenerate the expired values, you need to provide the ADMIN user credentials that you provided when you first switched to your own Oracle DB instance. Visual Builder uses the ADMIN user credentials to generate new Visual Builder tenant credentials to replace the expired credentials. Visual Builder does not store the ADMIN user credentials that you supply.

1. Open the General tab of the instance's Tenant Settings page.

If you cannot navigate to the Tenant Settings page from the navigation menu, you can open the page directly by entering the page's URL in the browser. The URL will be similar to

`https://<Instance-URL>/ic/builder/?root=settings&settingsSection=tenant-database.`

2. In the Tenant Database field, click the Edit icon to open the Update Tenant Database Connection wizard.

3. In the Update Tenant Database Connection wizard:
 - Reset expired credentials by supplying the ADMIN user credentials and ATP wallet that Visual Builder will use, or
 - Update the wallet for your Oracle DB instance by uploading the new ATP wallet.

Update Tenant Database Connection

Provide admin user credentials below to reset expired tenant schema password or ATP DB wallet.

Upload Wallet

Upload a zip file or drag one here

Wallet Password

DBA User Name Please make sure the user has DBA privileges.

Password The admin credential will be used to create or update schema only, and will not be saved.

Cancel Finish

4. Click **Finish**.

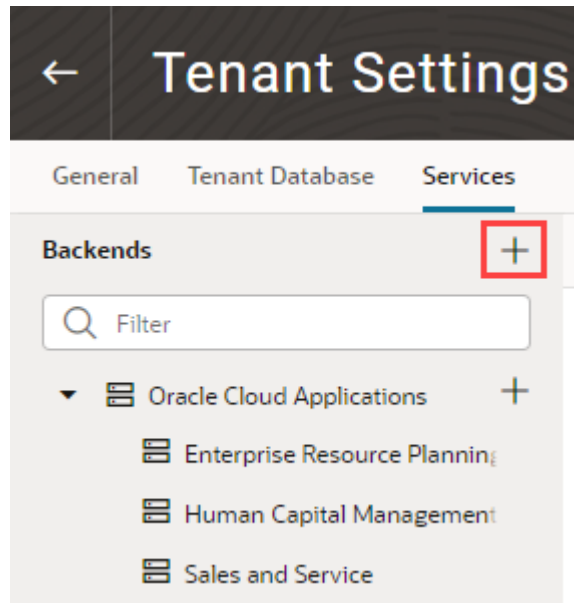
Add a Connection to Integration Applications

Administrators can use the Services tab in the Tenant Settings page to add a connection to an instance of Oracle Integration as a backend service.

If you are using multiple Visual Builder instances, for example, development and production instances, you might need to add connections to Oracle Integration in more than one instance.

To add a connection to an Oracle Integration instance:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend** and choose **Integrations** in the Create Backend dialog.



3. In the dialog, type the Server URL of the backend service, configure other settings such as security as needed, and click **Create**.

See About Authentication and Connection Type in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

Add a Connection to Oracle Cloud Applications

The list of REST services in the service catalog of a visual application is retrieved from an Oracle Cloud Applications backend service. Specify the instance URL of the Oracle Cloud Applications backend service in the Tenant Settings page.

All visual applications in the tenant will use the Oracle Cloud Applications instance URL specified in Tenant Settings, but a visual application can be configured to use a different Oracle Cloud Applications backend service by specifying a different instance URL in the Backends tab (which you access from the Navigator's Services tab). The tenant-level backend configuration is ignored if you or a visual application developer configures a different Oracle Cloud Applications backend service in a visual application's Backends tab.

The authentication choices available to configure a tenant-level Oracle Cloud Applications backend are:

- Basic Auth: Uses a fixed username and password for authentication.
- Oracle Cloud Account: Needs federation between Oracle Cloud Applications and Visual Builder.
- Delegate Authentication (previously called Propagate Current User Identity): Same as Oracle Cloud Applications. That is, it needs federation between Oracle Cloud Applications and Visual Builder.
- None: This assumes your Oracle Cloud Applications REST API can be called without any authentication, which is not usually the case.

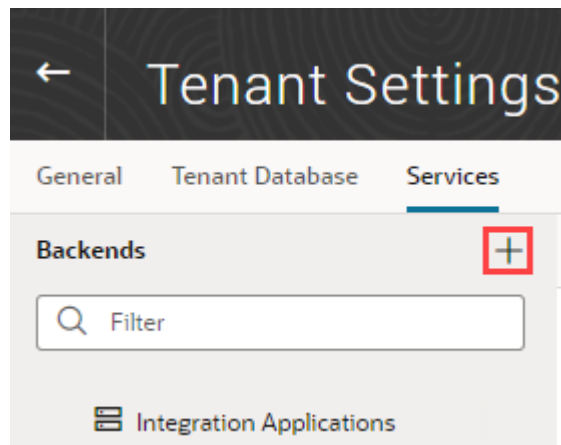
See About Authentication and Connection Type in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

If the necessary prerequisites for setting a tenant-level Oracle Cloud Applications backend service are not available, then a visual application developer can set up a backend service at

the visual application level where more options are available. Another option is for you (the service administrator) to configure the Oracle Cloud Applications backend with `None` and let the visual application developer override the authentication setting at the visual application level.

To specify an Oracle Cloud Applications service for the tenant:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend**, then choose **Oracle Cloud Applications** in the Create Backend dialog.



When specifying the URL in the Tenant Settings, you (the service administrator) only need to provide the instance URL of the Oracle Cloud Applications backend service to retrieve the list of services.

3. In the dialog, type the Server URL of the backend service, and configure other settings, such as security, as needed.
4. (Optional) After you configure settings for the backend, add headers to the backend.

Backend headers that you add will be applicable for any service connection to this backend, irrespective of the server or application profile that is used.

5. Click **Create**.

Visual Builder automatically discovers the `interfaceCatalogs` endpoint of the Oracle Cloud Applications backend, which retrieves the list of services and their metadata. This endpoint is typically in the form:

```
https://<My Oracle Cloud Applications Instance URL >/helpPortalApi/otherResources/latest/interfaceCatalogs
```

This endpoint is publicly accessible without any authentication.

If there is a problem creating the connection, verify the instance URL of the Oracle Cloud Applications instance.

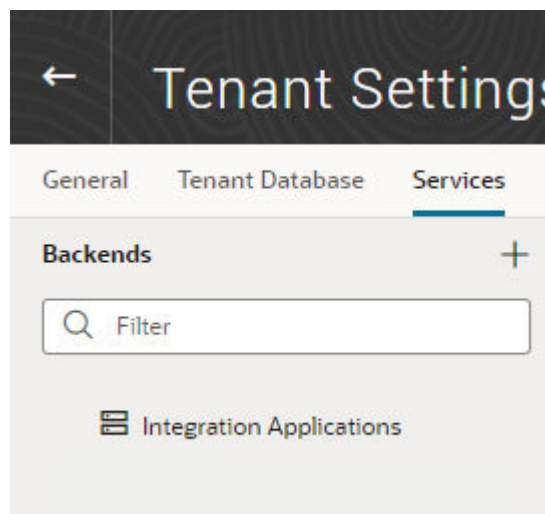
Add a Connection to Process Automation

Administrators can use the instance's Tenant Settings page to add a connection to an instance of Oracle Cloud Infrastructure Process Automation as a backend service.

If you are using multiple Visual Builder instances, for example, development and production instances, you might need to add connections to OCI Process Automation in more than one instance.

To add a connection to an OCI Process Automation instance:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend** and choose **Process Automation** in the Create Backend dialog.



3. Enter the URL of the instance, configure other settings, such as security, as needed, and click **Create**.

Add a Connection to Process Cloud Service

Administrators can use the instance's Tenant Settings page to add a connection to an instance of Oracle Process Cloud Service as a backend service.



Note:

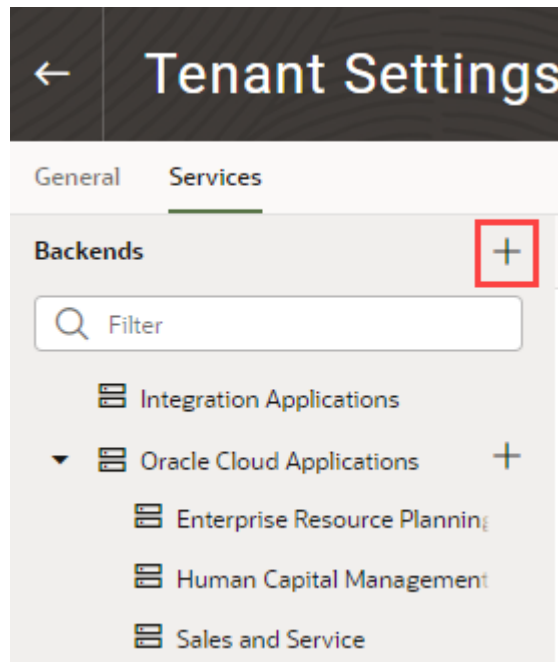
Oracle Process Cloud Service, which is included in the Enterprise edition of Oracle Integration Gen 2, is now deprecated; for details, see Process Features.

To add a connection to an instance of Oracle Process Cloud Service as a backend service, the instance of Oracle Process Cloud Service should be co-hosted with Visual Builder because the authentication types that Visual Builder supports for this configuration is Oracle Cloud Account or Propagate Current User Identity. In most cases, this backend service (Oracle Process Cloud Service) will be preconfigured for your Visual Builder instance.

If you are using multiple Visual Builder instances, for example, development and production instances, you might need to add connections to Oracle Process Cloud Service in more than one instance.

To add a connection to an Oracle Process Cloud Service instance:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend** and choose **Process** in the Create Backend dialog.



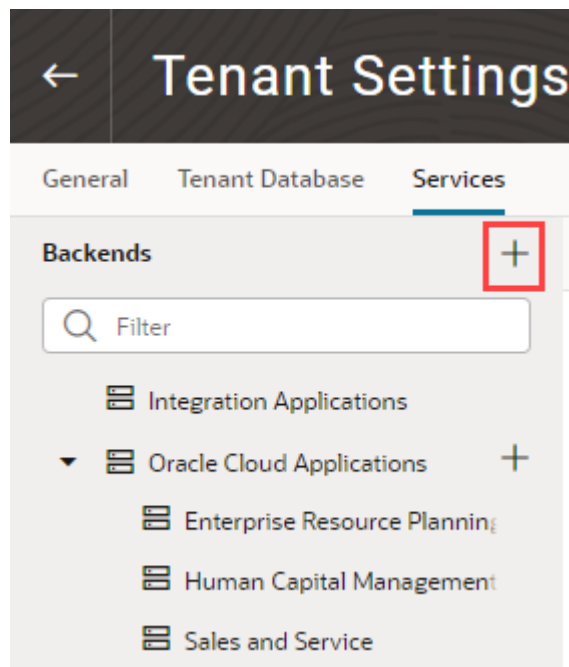
3. In the dialog, type the Server URL of the backend service, configure other settings, such as security, as needed, and click **Create**.

Add a Connection to a Custom Backend

You can create your own backend to map to a custom server other than the Oracle Integration, Process, and Oracle Cloud Applications backend services. You can create a custom backend with a free-form URL, or create a custom ADF backend when you know the Describe URL that points to an ADF Describe service.

To add a connection to a custom backend:

1. Open the instance's Tenant Settings page.
2. In the Services tab, click **Create Backend**.



3. In the Create Backend wizard, select the type of backend you want to create:
 - To create a backend with a free-form URL, click **Custom**.
 - To create a backend with the Describe URL of an ADF service, click **Custom ADF Describe**. Use this option only when your custom ADF Describe endpoint does not have any child backends.
4. In the Name field, enter a name and description for the custom backend.
5. Add headers to the backend. Backend headers that you add will be applicable for any service connection to this backend, irrespective of the server or application profile that is used.
6. Click **Next**.
7. Enter the instance URL for the custom backend, configure other settings, such as security, and click **Create**.

Create a Child Backend

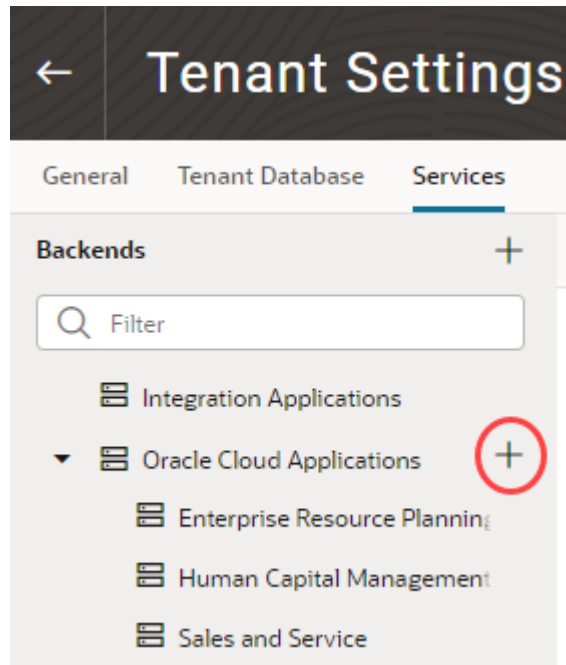
You can create child backends to extend the functionality provided by the top-level Oracle Cloud Applications or custom backend registered to your Visual Builder instance.

Let's say your instance's Oracle Cloud Applications backend connects to an Oracle Cloud Applications instance that provides access to these service catalogs: Enterprise Resource Planning Supply Chain, Sales and Service, and Human Capital Management. Now if you want to access another catalog (say, Search), you can create a child backend to access the search service.

A child backend inherits the parent backend's definition, which you can override as required. Its server URL is derived from the top-level backend, with `vb-catalog://backends/` as the base URL. Continuing the Oracle Cloud Applications example, the Sales and Service child backend adds to the top-level Oracle Cloud Applications backend and has `vb-catalog://backends/fa/crmRestApi/resources` as its server URL.


Child backends can be created only for the Oracle Cloud Applications backend and custom backends that use an OpenAPI/Swagger service specification.

- To create a child backend for the Oracle Cloud Applications backend:
 1. Open the instance's Tenant Settings page.
 2. In the Services tab, click the + sign for the top-level Oracle Cloud Applications backend:

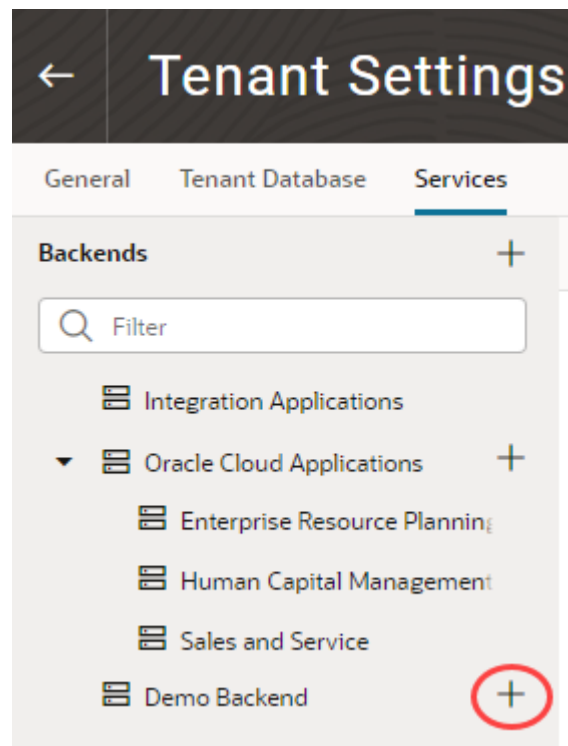


3. Select **Custom ADF Describe** to create a backend with an ADF Describe URL. For backends not having a Describe URL, select **Custom**.
4. Enter a name and description for the child backend. Optionally, add static headers.
5. Click **Next**.
6. Enter the instance URL for the child backend (for example, `vb-catalog://backends/fa/applcoreApi/search/`). The child backend's URL will usually start with `vb-catalog://backends/oracle-cloud-app-BackendId`.

 **Tip:**

To see the complete URL that the backend resolves to, click the **Detach** icon ()

7. Enter other settings, such as security and headers.
 8. Click **Create**.
- To create a child backend for a top-level custom backend:
 1. Open the instance's Tenant Settings page.
 2. In the Services tab, click the + sign for a top-level custom backend:



3. Enter a name and description for the child backend. Optionally, add static headers.
4. Click **Next**.
5. Enter the instance URL for the child backend (for example, `vb-catalog://backends/demo/newdemo`). You can click the **Detach** icon to see the complete URL that the backend resolves to.
6. Enter other settings, such as security and headers.
7. Click **Create**.

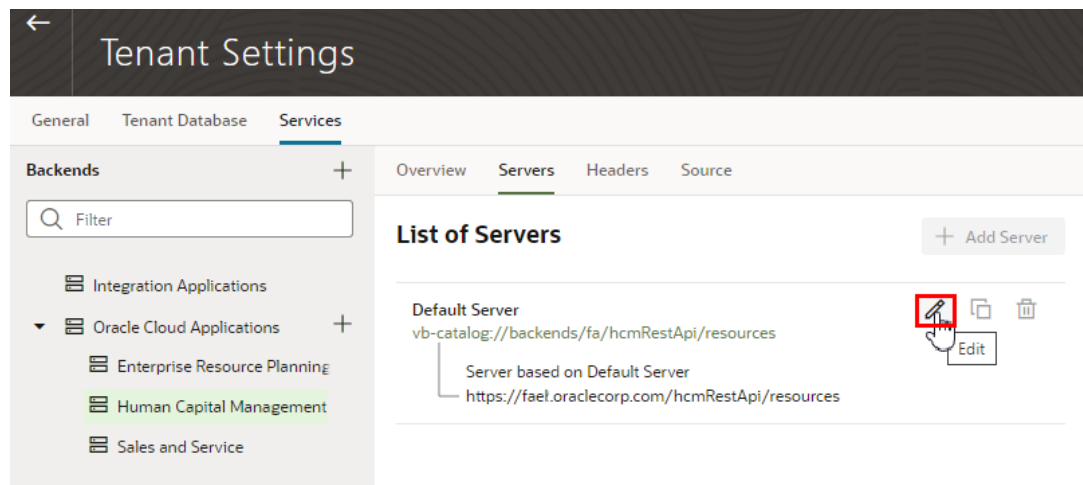
Edit Authentication for a Backend Service

Administrators can use the Services tab in the Tenant Settings page to edit the authentication and connection settings for the backend services available in the tenancy. Once a backend service is added, you can edit its details. In the case of child backends, you can also override the settings inherited from the backend service, for example, to allow connections to the child HCM backend to use basic authentication instead of using the Oracle Cloud Account authentication for logged-in users set at its parent backend.

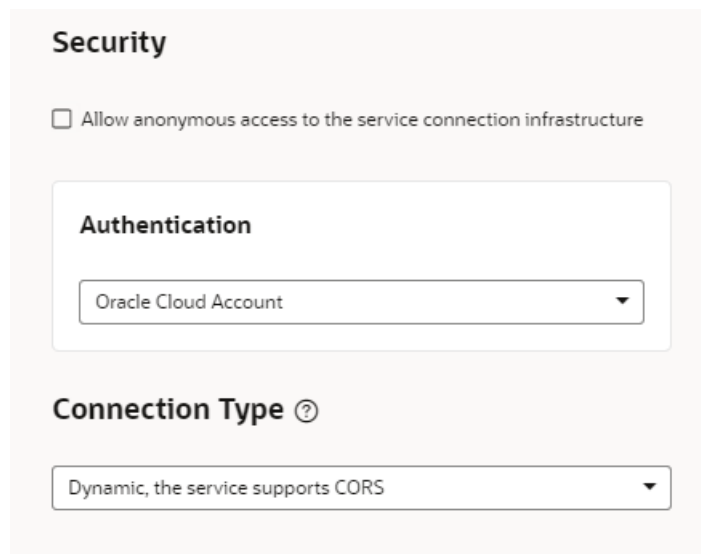
For a description of the authentication options, see *About Authentication and Connection Type* in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

To edit the connection details for a backend service:

1. Open the instance's Tenant Settings page.
2. In the Services tab, select the backend you want to edit.
3. Open the Servers tab of your backend, and then click **Edit**.



4. In the Edit Server dialog box, edit the settings in the Security and Connection Type panes.



In the case of a child backend, the authentication and connection type details for your backend service are inherited from the parent service, so you'll need to override the settings. If you want to revert your changes for a child backend, you can click **Return to inherited** to restore the default inherited setting.

Edit Server [Close]

Server Identification

Instance URL ⓘ
vb-catalog://backends/ta/hcmRestApi/resources

Description Keep it short so you can identify the server easily
Default Server

Server Variables

URI Template expressions entered in the URL field will be listed here.

Headers

Custom Headers
+ Add Header

Secure Headers
+ Add Header

Security Inherited from Backend

Allow anonymous access to the service connection infrastructure

Authentication for Logged-In Users
Authentication: Oracle Cloud Account
Override security

Connection Type ⓘ Inherited from Backend

Always use proxy, irrespective of CORS support
Override Connection Type

Cancel Save

5. In the Security pane, select an authentication type in the dropdown list.
If you are editing a child backend, you'll click **Override security** in the Security pane, and then select an authentication type in the dropdown list.

Security

Allow anonymous access to the service connection infrastructure

Authentication for Logged-In Users

Oracle Cloud Account

None
No authorization headers. Use when the service requires no authentication or when you want to pass a custom authorization header

Oracle Cloud Account
Logged in user identity represented by OAuth token with URL as scope

Delegate Authentication
Authentication determined by the calling web or mobile app's security settings. (For requests initiated from Service Tester, the logged-in developer's identity is used.)

OAuth 2.0 Client Credentials
OAuth token obtained with fixed client id and secret

OAuth 2.0 Resource Owner Password Credentials
OAuth token obtained with fixed client id, secret, username and password

OAuth 2.0 User Assertion
Logged in user identity represented by OAuth token for custom scope

Basic
Fixed username and password

Oracle Cloud Infrastructure API Signature 1.0

Cancel Save

6. In the Connection Type pane, select a connection type in the dropdown list.
In the case of a child backend, click **Override Connection Type** and then select the connection type.
7. Click **Save**.

Manage Self-signed Certificates

Administrators can use the Certificates page to upload and manage the self-signed certificates used by the instance to enable inbound and outbound SSL communications to a service's REST APIs

When creating connections to REST services that use self-signed certificates, you might need to add an API's certificate to your Visual Builder instance to validate SSL connections to that service. You can use the Certificates page to upload and remove certificate files (.pem) for


services. Uploading a service's certificate file to the keystore will allow all applications in the instance to communicate with that service. The Certificates page displays a list of certificates that have been added. You can click the Delete button in a row to remove the certificate.

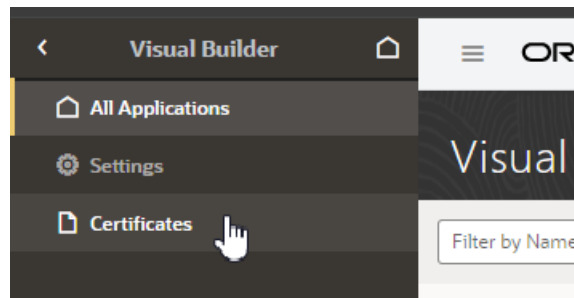
 **Note:**

Your staged or published apps might stop working if they use service connections with self-signed certificates and the certificates have expired. Any certificates issued after 2020-09-01T00:00:00.00Z will automatically expire 398 days after they have been issued. If your apps use certificates issued before 2020-09-01T00:00:00.00Z, the certificates will not expire, but you should update them with a newer certificate.

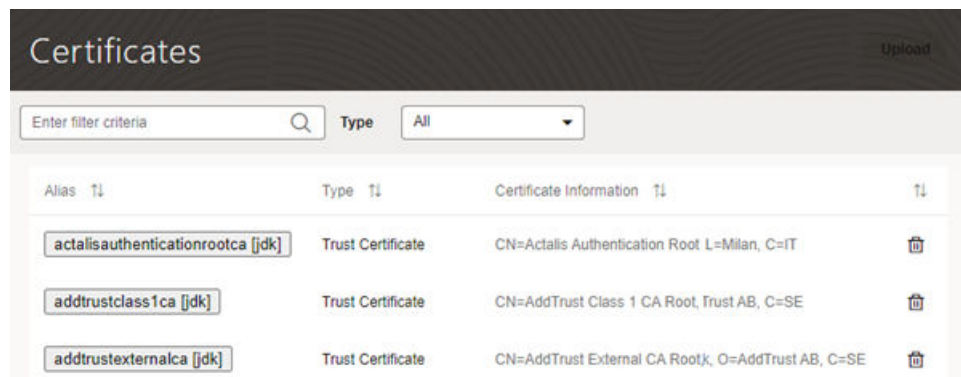
To avoid disruptions, you should plan regular updates to refresh the self-signed certificates before they expire (for example, every 6 months). It's not recommended to use self-signed certificates in production apps.

To upload a self-signed certificate:

1. In the upper-left corner of the Visual Builder title bar, click **Navigation Menu** .
2. Click **Certificates** in the navigation menu to open the Certificates page.



The Certificates page displays a list of the certificates already uploaded to the instance.



3. Click **Upload** to open the Upload Certificate dialog box.

You use the Upload Certificate dialog box to create an alias for the certificate and upload the service's certificate file from your local system.

Upload Certificate ×

Enter a certificate alias name, select certificate type, and pick a certificate file. Certificate allows integration Cloud Service connect with external services. If the external service/endpoint needs a specific certificate, request the certificate and then imports it into Integration Cloud Service.

Certificate Alias Name *

Certificate Type *

Certificate *

Drag and Drop
Upload or drag certificate file here

4. Type the alias in the Certificate Alias Name field.
The alias is used to identify the certificate in the table in the Certificates page. The Certificate Type dropdown list is read-only because only Trust Certificates are supported.
5. Drag the certificate file from your local system into the upload target area, or click the upload target area to browse your local system.
6. Click **Upload** to add the certificate to the service keystore.

Manage Your Component Exchange

If your team develops custom components for visual applications and want the components to be available to all users in the Visual Builder Components tab, you'll need to first set up a component exchange. This chapter tells you how to set up the Component Exchange in Visual Builder.

What is a Component Exchange?

A component exchange is a repository of custom components available in VB Studio. You can use these components in your visual applications, such as web components and application templates. Many of the components provided by Oracle can be installed from a component exchange.

To integrate a component exchange with a Visual Builder instance, you provide the exchange's URL and credentials in the Tenant Settings. The exchange can be a private exchange in a VB Studio project or one of the exchanges maintained by Oracle.

If your organization develops or uses proprietary components, these components can be published to a private exchange hosted by a VB Studio project. For example, if you have a web component designed to be used in applications in your tenant, you can set up your own exchange and use it to distribute the component to developers in the tenant. Additionally, components provided by Oracle are automatically available from all private component exchanges.

Oracle maintains two component exchanges containing components validated by Oracle that are publicly available to all developers. If you don't have a private exchange but you want to

give developers access to these Oracle components, you can add one of the following exchanges maintained by Oracle. If your instance is in the US, use the following details.

| Field | Value |
|-------------|---|
| Service URL | <code>https://component-exchange-soctesting2-phx.developer.ocp.oraclecloud.com/component-exchange-soctesting2-phx/s/component-exchange-soctesting2-phx_compcatalog_30314/compcatalog/0.2.0</code> |
| Username | <code>compcatalog.user</code> |
| Password | <code>k9fz-0Pw4x-q</code> |

If your instance is in Europe, use the following details.

| Field | Value |
|-------------|---|
| Service URL | <code>https://component-exchange-soctesting4-fra.developer.ocp.oraclecloud.com/component-exchange-soctesting4-fra/s/component-exchange-soctesting4-fra_compcatalog_11494/compcatalog/0.2.0</code> |
| Username | <code>compcatalog.user</code> |
| Password | <code>k9fz-0Pw4x-q</code> |

About Component Exchanges Hosted in VB Studio Projects

A VB Studio project can host a secure component exchange to store and distribute components only available to developers in the instance.

Each VB Studio project includes the component exchange 'compcatalog', which is the service used to access components stored in the project. The compcatalog service is provisioned by default with each project. Any project can be used to host an exchange if storage is enabled for the VB Studio instance. Component developers can use the service's APIs to publish components to the exchange.

To integrate a private exchange in a VB Studio project with a Visual Builder instance, an administrator specifies the URL for the project's compcatalog service and the credentials for a user that can access the project. The credentials used to connect to the exchange must be an owner or member of the VB Studio project hosting the exchange. All developers in the tenant use these credentials to connect to the exchange to get the components and application templates they want to use in their projects.

The URL for the project's compcatalog service has the following form: `https://<hostname>/<org_id>/s/<project_id>/compcatalog/0.2.0/`

In the URL, "compcatalog" is the exchange service and "0.2.0" is the API version of the service.

To determine the URL for the compcatalog service, you need to know the following details about the VB Studio project:

- `<hostname>`. This is the VB Studio server where the project is hosted.
- `<org_id>`. This is the organization (tenant) name.
- `<project_id>`. This is a project identifier unique to the tenant. This is not the same as the project display name entered by the project owner and is not displayed in the VB Studio UI.

If you do not know the `<project_id>` for the project hosting the exchange, you can get it from the Git or Maven configuration, or by using the VB Studio Projects API. The following table describes how to get the `<project_id>`.

| Method | Steps |
|-----------------------------------|---|
| From a Git or Maven configuration | <ol style="list-style-type: none">1. In VB Studio, open the project and locate the Repositories tab on the project's Home Page.2. Expand the the Git or Maven section and copy the repository URL. <p>The Git repository URL will be similar to the following: <code>https://{user_id}@{hostname}/{org_id}/s/my-org_testproject_5/scm/my-repo.git</code></p> <p>The Maven repository URL will be similar to the following: <code>http://{hostname}/{org_id}/s/my-org_testproject_5/maven/</code></p> <p>In these examples, "my-org_testproject_5" is the project identifier. In this case, the URL for the 'compcatalog' service will be similar to <code>https://{hostname}/my-org/s/my-org_testproject_5/compcatalog/0.2.0/</code></p> |
| Using VB Studio Projects API | <p>If you know the name of the project sharing your exchange instance, you can get the project metadata using a REST call to the VB Studio API.</p> <p>For example, you can use cURL to send a REST call similar to the following:</p> <pre>curl -X GET -u '{username}:{password}'https://{hostname}/{org_id}/api/v2/projects/info/name:TestProject</pre> <p>The return should be similar to the following:</p> <pre>[{ "organization":"my-org", "identifier":"my-org_testproject_5", "name":"TestProject", "urlId":"testproject", "description":null, "accessibility":"PRIVATE", "template":false, "state":"READY", "locked":false, "relation": {"membership":"OWNER","favorite":false} }]</pre> <p>In this example, the identifier property in the return is the project identifier that is needed for the "compcatalog" service URL.</p> |

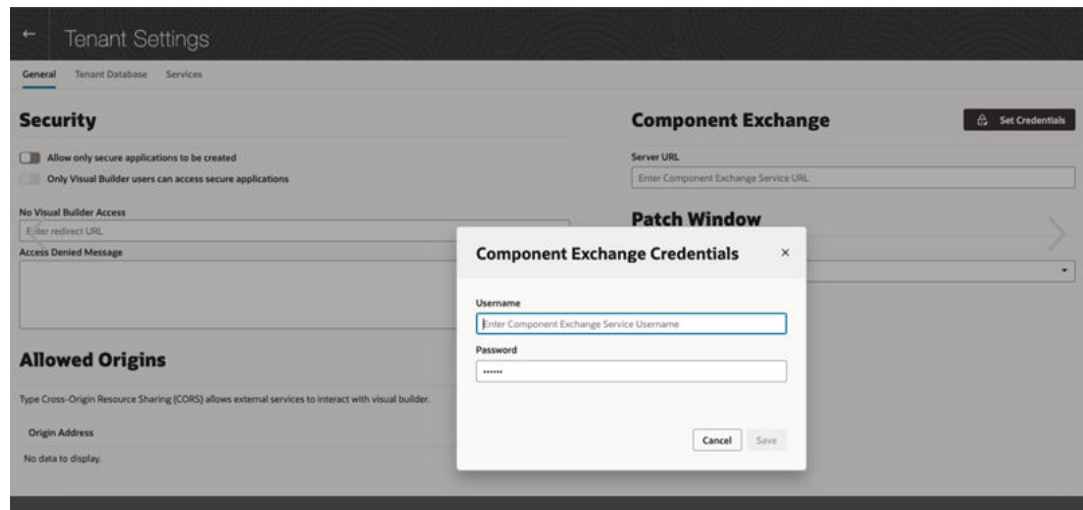
Add a Connection to a Component Exchange

When an instance is integrated with a component exchange, all developers using the instance can access and install components stored there.

After an exchange is added to the instance, all developers can use the Components tab in the Navigator to install and manage the components from the exchange that they want to use in their applications. When creating an application in the Create Application wizard, developers can also select any of the application templates that have been published to the exchange.

To add a connection to the Component Exchange:

1. Open the instance's Tenant Settings page.
2. In the Component Exchange panel, enter the URL for the component exchange, then click **Set Credentials**.
3. In the Component Exchange Credentials dialog box, enter the username and password for the user whose credentials will be used to authenticate access to the Component Exchange. Click **Save**.



If you are adding a connection to a private component exchange, it is recommended that the credentials you provide are for an administrator who is a member of the VB Studio project hosting the exchange or the project owner.

Configure Support for a Custom Domain

If you want your customers to see a different URL than the one generated by Visual Builder, you can map a custom domain to your instance. A custom domain is a customer-provided hostname and domain (FQDN) created by adding a subdomain to your domain. After configuring your instance to use a custom domain, app users accessing the app using the custom domain will not see the typical Oracle domain (for example, `myvbinstance-accountname.builder.ocp.oraclecloud.com`) in the URL, but instead would see something like `mycustom.example.org`.

To use a custom domain for a visual application:

- Configure the custom URL for your Oracle Integration 3 instance, using the instructions in [Create and Configure a Custom Endpoint for Your Visual Builder Instance](#).

 **Note:**

Custom URLs are supported on Oracle Integration instances, as well as on Visual Builder instances.

Depending on whether you're on Oracle Integration, Visual Builder, or a Visual Builder instance that was provisioned as part of a SaaS order, the process for enabling custom URLs varies. If you are using an Visual Builder instance, use the instructions in Configure Support for a Custom Domain in *Administering Oracle Visual Builder Generation 2*.

- Set the custom domain in the visual application's Settings editor and publish the app. See Specify a Custom App URL in *Developing Applications with Oracle Visual Builder in Oracle Integration 3*.

 **Note:**

You can configure only one custom domain for each Oracle Integration 3 instance.

After configuring a custom domain:

- Users can access a single web app by typing just the custom domain URL in the browser, for example, `mycustom.example.org`. The app is loaded from the custom domain root ("/"), and no additional path information or query parameters are required in the URL.
- `http` can be redirected to `https`, so if a user types "mycustom.example.com", this will resolve to `https://mycustom.example.com`, and load the default web app.
- For applications that contain business objects, the Business Object REST API can also use the custom domain configuration.
- Developers can access the Designer in Visual Builder using a custom domain.
- If you create and stage an application from a custom domain (`https://mycustom.example.com/ic/builder/designer`), you'll be automatically redirected to the custom domain (`https://mycustom.example.com/ic/builder/rt/appid/version/...`) when you open the app using a URL that isn't the application's custom domain (for example, your instance's URL `https://servicename.oraclecloud.com/ic/builder/rt/appid/version/...`).

Custom domains are also subject to other limitations:

- A custom domain can only access a published app. It will not work for apps that are only staged.
- If you publish a different web app in your visual application, it immediately becomes the default app for the custom domain, and the previous web app will no longer be available at the custom domain.
- A custom domain can only be used to access one live app (in the visual application configured for the root URL). You can access other live apps in the same instance only by using the full Oracle Cloud URL or by creating and configuring a different custom domain and visual application.
- If a visual application contains more than one web app, only one of them can be accessed using the custom domain. It's not possible to specify which app in a visual application will

be available at the custom domain because the domain is configured in the Settings for the visual application, not for individual web apps. If you are going to use a custom domain, it is recommended that the visual application only contain one web app to ensure that the correct app is loaded.

- Mobile and PWA apps are not supported at this time. Custom domains can only be used for web apps.