Release Manager Document

Version No: 1.0

Approved By: Vahana Team

Decimal Technologies

8th Floor, Tower D Pioneer Urban Square,
Golf Course Ext Rd, Sector 62, Gurugram, Haryana-122102

Document Control

Document Control

S.No	Type of Information	Document Data
1.	Title	Release Manager Document
2.	Document No	RMD_01
3.	Document Version No	1.0
4.	Document Owner	Yogesh Sharma
5.	Document Author(s)	Vikas Dhillon/Ankita Lakra/Kumar Saurabh
6.	Document Approver	Vahana Team

Document Update Summary

Version No	Revision Date	Nature of Change	Reviewer	Date Approved
V 1.0	15/Mar/2022	1st Draft	Ankita Lakra	N/A

Table of Contents

Release Manager Document

1. Introduction

1.1 Document Purpose

The purpose of this document is to explain the function of the Release Manager application. The document includes the overview of Release Manager and describes how a user can use the Release Manager application in the Vahana cloud.

The Release Manager application helps you manage multiple versions of the Vahana platform and the applications/services that are deployed into the Vahana cloud.

1.2 Document Scope

The functional scope of this document includes the following sections:
Section1:- This is the current section of the document, which contains:
The objective and purpose of the document
Functional scope of the document
The user group/audience that can access the document, and others

Section2:- This section contains the overview and definition of the Release Manager application and the broad description of all the modules that are incorporated into Release Manager.

1.3 Intended Audience

Different members of the Vahana cloud development and quality analysis teams can access this document to capture the functional details of the Release Manager application.

1.4 Acronyms and Abbreviation

The following table contains the list of abbreviated terms that are repeatedly used in the current document. The table also contains the description of these abbreviated terms.

Term	Description
UAT	User Acceptance Testing
VRT	Vahana Run Time
JSON	Java Script Object Notation

1.5 Reference Document

For the Release Manager application, this document does not refer to Decimal proprietary or any other third party document.

Document Name	Version	Date	Company/Organization
-	-	-	-

2. Release Manager Overview

Release Manager is a web based application that is integrated with Vahana 2.0 cloud platform. Based on the concept of GITHub, Release Manager is a code hosting and the version controlling and management application. It helps you work and manage multiple versions of the application/service or module.

Prior to the development of the Release Manager application, the Vahana platform development team had to push the code from the Production environment to the UAT (User Acceptance Testing) environment and then perform the code-fix or error fixing. After the code was fixed or changed, the developer had to merge the code back to the production environment.

This code migration from production environment to UAT environment or vice versa was tedious and time-consuming. This also tended to compromise the application up-time.

To remove these hiccups, the Vahana development team developed and launched the Release Manager application. The Release Manager application smartly allows you to cut a new child branch out of the master branch. Release Manager consists of a default master branch, which is called the "Development" branch. All the new branches are cut out of the "Development" branch. Therefore, you can change or fix the code in the new child branch and then merge back the child branch into the "Development" branch.

In the Release Manager application, you can create three types of child branches as follows:

- **Hotfix branch**
- **Deployment branch**
- **Backup branch**

The hot-fix and deployment categories are mainly used to fix and change the application code. After the code is fixed and changed, it is synced on the VRT. The backup category is used to take the backup of the code from the master branch. You can create new child branches from the master branch up to an "n" number of hierarchies.

This advantage gives you the leverage and freedom to work on the application code in the new branch without touching the application code in the production branch. In case, if the application code that you work on in the new child branch does not work properly, you can perform the roll-back by discarding the changes in the child branch.

2.1 Release Manager Features

The Release Manager application offers the following features:

➢ **Release Management**

This module allows you to:

- Deploy the branch
- Roll-back the branch
- Sync the changes of the hotfix/deployment branch
- Access the details of the branch

➢ **Branch Management**

This module allows you to:

- Create a new branch from the default source branch
- Access the details of a hot-fix branch, including
  - **Name of the hot-fix branch**
  - **Type of the hot-fix branch**
  - **Name of the source branch from which the hot-fix branch is created**
  - **Deployment details, and others**
- Edit the details of the hot-fix branch

➢ **Deployment History**

This module allows you to:

- Access the deployment details on the basis of the type and the name of the environment
- Access the deployment details on the basis of the type and the name of the hot-fix branch

➢ **View Sync Status**

This module allows you to:

- View the sync status on the basis of the name of the environment
- View the sync status on the basis of the application module in which a component is synced or is pending to be synced
- View the configuration details of the synced component in JSON format

3. Release Manager Modules

The Release Manager application consists of the following modules, which are broadly described as below:

3.1 Release Management

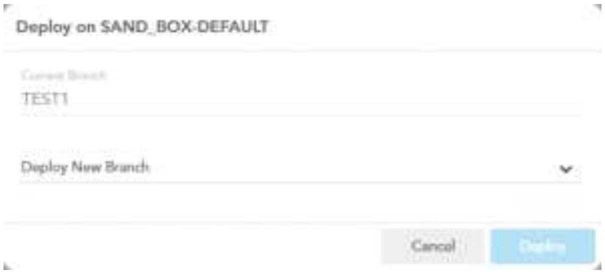
This feature allows you to perform several functions such as sync the changes, deploy or roll back the branch, view the details of the branch, etc. You can perform these functions as follows:

3.1.1 Deploying a Branch

When you initiate the deployment of a branch, the branch is automatically deployed on VRT (Vahana Run Time). On the Vahana cloud platform, VRT is a component that allows you to allocate a virtual space to the newly created environment.

To deploy a branch on VRT:

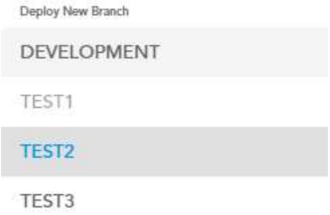
1. In the **Release Manager** application, click the **Release Management** tab, the application displays the environment details.
2. In the **Default** area, click the **Deploy Branch** icon (✎), the **Deploy on <<Environment Name>>** dialog box opens.
3. In the dialog box, click the **Deploy New Branch** list, the application displays the list of the branches that are not deployed.
4. In the list, select the branch that you want to deploy.
5. After you select the branch, click **Deploy**, the branch is successfully deployed on VRT.



Deploy on SAND_BOX-DEFAULT

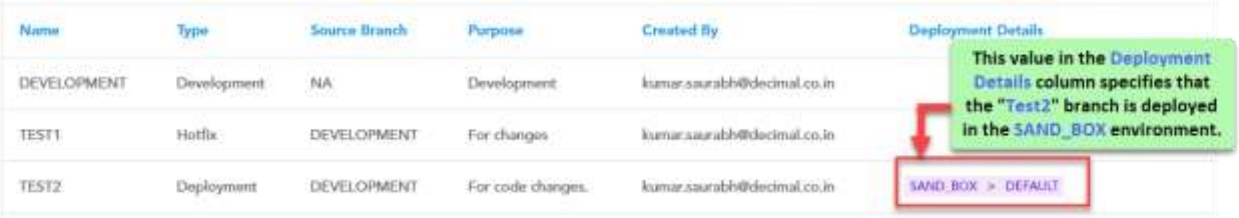
Current Branch
TEST1

Deploy New Branch



Deploy New Branch

- DEVELOPMENT
- TEST1
- TEST2
- TEST3



Name	Type	Source Branch	Purpose	Created By	Deployment Details
DEVELOPMENT	Development	NA	Development	kumar.saurabh@decimal.co.in	
TEST1	Hotfix	DEVELOPMENT	For changes	kumar.saurabh@decimal.co.in	
TEST2	Deployment	DEVELOPMENT	For code changes.	kumar.saurabh@decimal.co.in	SAND_BOX > DEFAULT

This value in the Deployment Details column specifies that the "Test2" branch is deployed in the SAND_BOX environment.

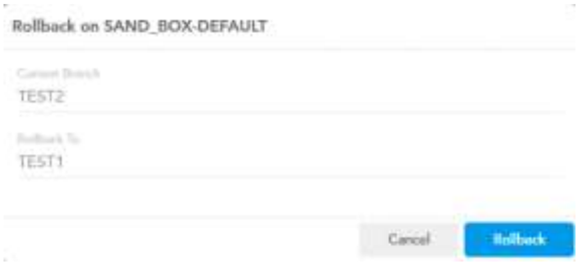
3.1.2 Deployment Roll-back

This functionality allows you to roll back the deployment of the earlier deployed branch. You can roll back the deployed branch as a result of any of the following reasons:

- Erroneous code in the deployed branch or,
- After the branch is deployed, the deployed code halts the application's function

To roll back the deployed branch:

1. In the **Release Manager** application, click the **Release Management** tab, the application displays the environment details.
2. In the **Default** area, click the **Rollback Branch** icon (↺), the Rollback on <<Environment name>> dialog box opens.
3. The dialog box displays the following details:



Rollback on SAND_BOX-DEFAULT

Current Branch
TEST2

Rollback To
TEST1

Box	Description
Current Branch	This box displays the name of the currently deployed branch.
Rollback To	This box displays the name of the branch that was deployed into the environment before the current branch is deployed. After you perform the rollback, the branch that the **Rollback To** box displays will be redeployed.

4. On the **Rollback on** dialog box, click **Rollback**, the application rolls back the currently deployed branch and then reverts the deployment to the last deployed branch.

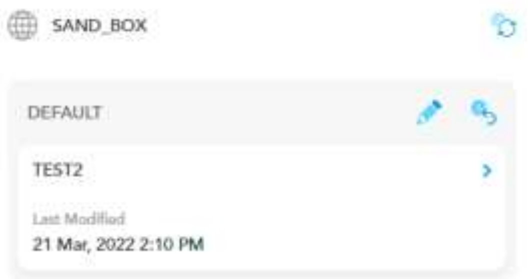
3.1.3 Syncing Changes

This feature allows you to sync the changes of the hot-fix or the deployment branch on VRT. Syncing the changes specifies that you are imparting the changes of the deployment branch/hot-fix branch to the Vahana Run Time environment.

After you update a module or rectify the application code of the module, it is necessary to sync the updated or rectified code with the code that is running on the Vahana Run Time environment.

To sync the changes:

1. In the **Release Manager** application, click the **Release Management** tab, the application displays the environment details.
2. In the details of the environment, click the **Sync <<Environment Name>>** icon (🔄), the changes of the currently deployed branch are synced on VRT.



Last Modified
21 Mar, 2022 2:10 PM

3.1.4 Accessing Branch Details

In the Release Management module, this feature allows you to access the details of the branch. The application displays the details of the branch at the component level as follows:

- **List of components that you access in the hot-fix or deployment branch**
- **Type of component**
- **Version of the component and,**
- **A feature to access the configuration details of the component**

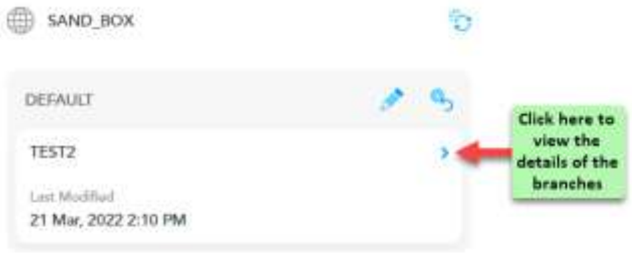
After you cut a branch out of the master branch to access a component, the application maintains the details of the branch on the basis of the component ID. The component ID displays the name of the component that you access in the branch.

Suppose, you cut a branch to access the same component more than once. Each time you access the component, you impart changes in the component and then deploy it. In this case, the Release Management application will maintain the updated component with the incremental version number.

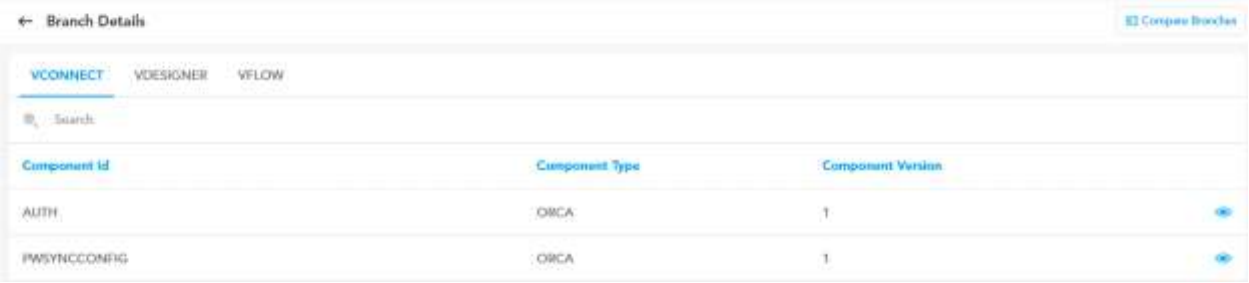
You can view the list of components application-wise. This feature also allows you to view the configuration details of the component in the JSON (Java Script Object Notation) format.

To access the branch details:

1. In the **Release Manager** application, click the **Release Management** tab, the application displays the environment details.



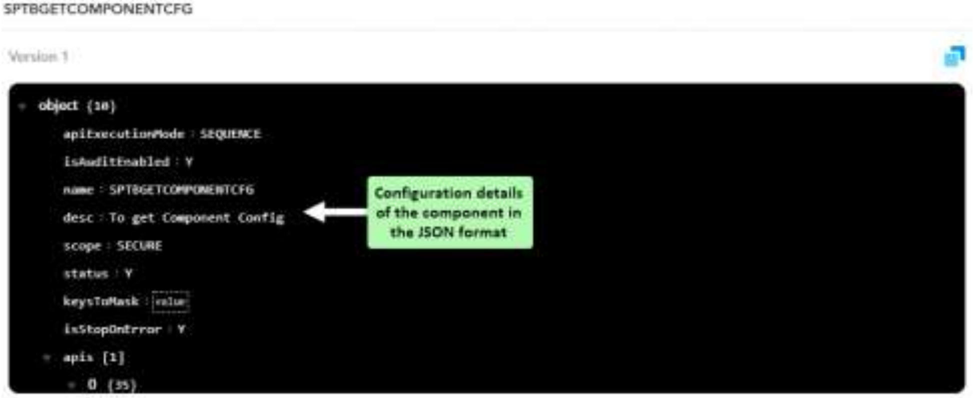
2. In the **Default** area, click the horizontal arrow icon (›), the **Branch Details** page opens.



Component Id	Component Type	Component Version
AUTH	ORCA	1
PWSYNCCONFIG	ORCA	1

3. The **Branch Details** page displays the list of components based on the application name and type.
4. Click the tab of the application name (For example: - vConnect or vDesigner), the application displays the list of the components that have been updated or fixed.
5. The **Branch Details** page displays the following details at the component level:

Branch Detail	Description
Component ID	This column displays the name of the component that you access in the hot-fix or deployment branch.
Component Type	This column displays the type of the component that you access in the hot-fix branch. The Release Manager application maintains different types of components at the system level. These component types are named variables or keywords. Based on the component that you access in the hot-fix/deployment branch, the application assigns a predefined type to the component.
Component Version	This column displays the version number of the component. The Release Manager application maintains the version of the specific updated component incrementally.
Eye Icon (👁)	If you click the eye icon (👁), the application displays the configuration details of the component in the JSON format.

	 SPTBGETCOMPONENTCFG Version 1 object {10} apiExecutionMode : SEQUENCE isAuditEnabled : Y name : SPTBGETCOMPONENTCFG desc : To get Component Config scope : SECURE status : Y keysToMask : value isStopOnError : Y apis [1] 0 {35} Configuration details of the component in the JSON format

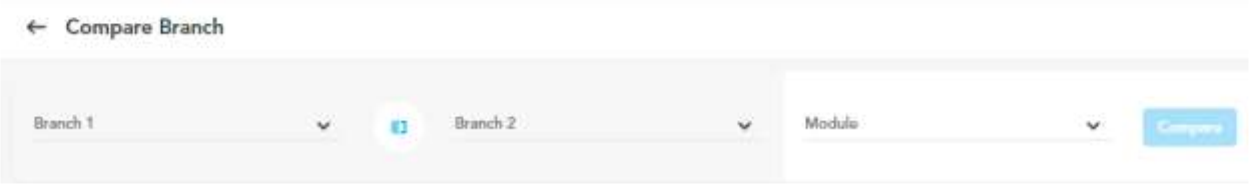
3.1.5 Comparing Branches

This feature allows you to compare two branches. While comparing two different branches, you can view:

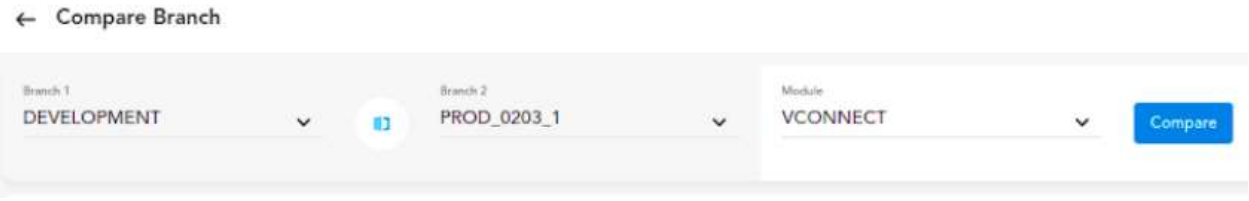
- **The name of the component in the branch**
- **Version number of the component, etc.**

You can use the “Compare Branches” feature to compare two different versions of the same component. To compare two different branches:

1. In the Release Management application, locate the top-right area and then click **Compare Branches**, the **Compare Branch** page opens.

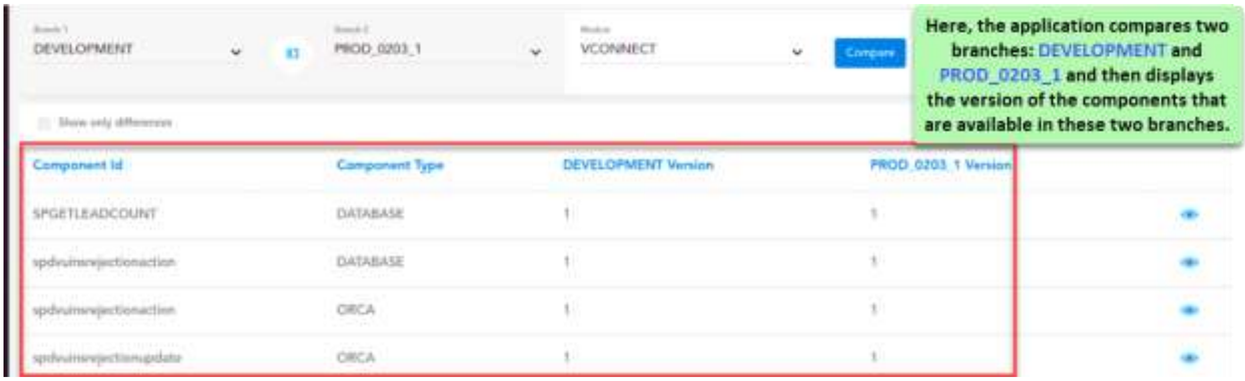


2. On the **Compare Branch** page:
   - Click the **Branch 1** list and then select the first branch (For example: - DEVELOPMENT) to compare.
   - Click the **Branch 2** list and then select the second branch (For example: - PROD_0203_1) to compare.
   - Click the **Module** list and then select the application (For example: - vConnect).



Branch 1: DEVELOPMENT
Branch 2: PROD_0203_1
Module: VCONNECT

3. After you select the branches and the application module, click **Compare**, the application displays the following details after it compares two branches:
   - **Name of the component under Component ID**
   - **Type of component (For example:- Database)**
   - **Version of the component in the first branch**
   - **Version of the component in the second branch**

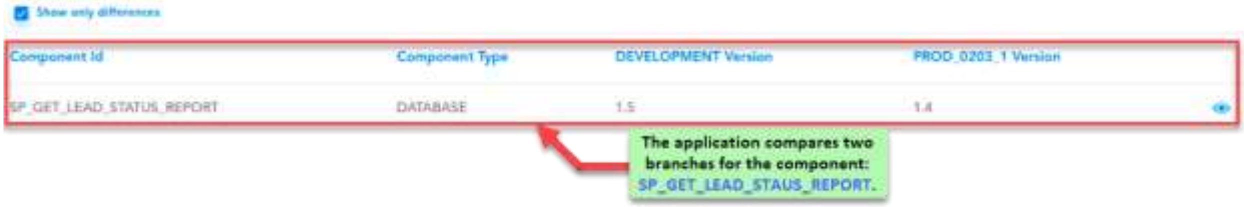


□ Show only differences

Component Id	Component Type	DEVELOPMENT Version	PROD_0203_1 Version
SPGETLEADCOUNT	DATABASE	1	1
spdvuinsrejectionaction	DATABASE	1	1
spdvuinsrejectionaction	ORCA	1	1
spdvuinsrejectionupdate	ORCA	1	1

Here, the application compares two branches: DEVELOPMENT and PROD_0203_1 and then displays the version of the components that are available in these two branches.

4. On the **Compare Branch** page, click the **Show only differences** check box, the application displays the list of the components that the Release Manager application maintains with multiple versions.



☑ Show only differences

Component Id	Component Type	DEVELOPMENT Version	PROD_0203_1 Version
SP_GET_LEAD_STATUS_REPORT	DATABASE	1.5	1.4

The application compares two branches for the component: SP_GET_LEAD_STAUS_REPORT.

3.2 Branch Management

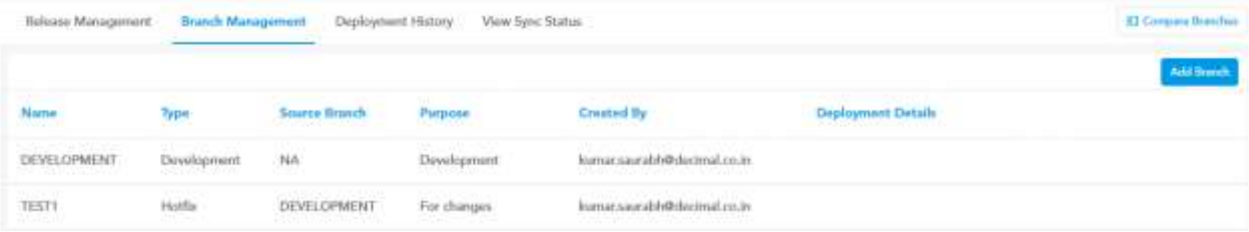
This feature allows you to create, manage, and edit the branches. In the Release Manager application, a branch is created out of the master branch. As described in the heading section: Release Manager Overview, you create three types of branches from the master branch: Hotfix branch, Deployment branch, and backup branch. The backup is used to archive the application code. The deployment and hotfix branch are commonly created to access and update/fix the application code.

After the code is fixed, the hotfix or deployment branch is merged back into the master branch. You create, manage, and edit a child (hotfix, deployment) branch as follows:

3.2.1 Creating a New Branch

This feature allows you to create a new branch. To create a new branch:

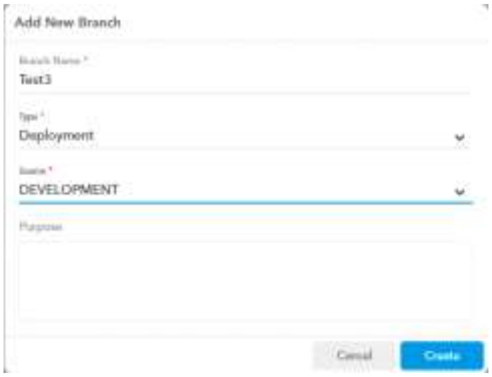
1. In the **Release Manager** application, click the **Branch Management** tab.



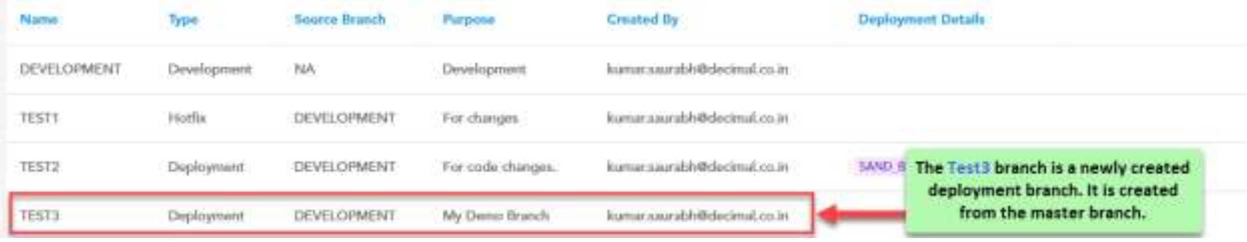
Name	Type	Source Branch	Purpose	Created By	Deployment Details
DEVELOPMENT	Development	NA	Development	kumar.saurabh@decimal.co.in	
TEST1	Hotfix	DEVELOPMENT	For changes	kumar.saurabh@decimal.co.in	

2. Click **Add Branch**, the **Add New Branch** dialog box opens.
3. On the **Add New Branch** dialog box, perform the following functions:

Box/List	Description
Branch Name	In this box, enter the name of the branch (For example: - Test3).
Type	In this box, select a category or type for the branch as follows: ➢ **Deployment** Select this type if you want to access the application module to fix or update the application code. ➢ **Backup** Select this type if you want to take the backup of the application code. ➢ **Hotfix** Select this type if you want to access and then fix the application code to resolve the abruptly occurring errors.
Source	Click this list and then select the source branch from which you want to create a new branch. You can select the source branch as follows: ➢ **Select the master branch** Select the master branch if you want to create a new branch from the master branch. After you fix or update the application code, the child branch is merged back into the master branch. ➢ **Select the child branch** You can also create a new branch from a non-master branch. In this case, the

	source branch can be a deployment or hotfix, or a backup branch. It means that you can create an 'n' number of child branches from another child branch.
Purpose	In this box, enter the brief purpose to create the new branch.  Add New Branch Branch Name *: Test3 Type *: Deployment Source *: DEVELOPMENT Purpose

4. After you enter or select the data to create a new branch, click **Create**, the new branch is created.



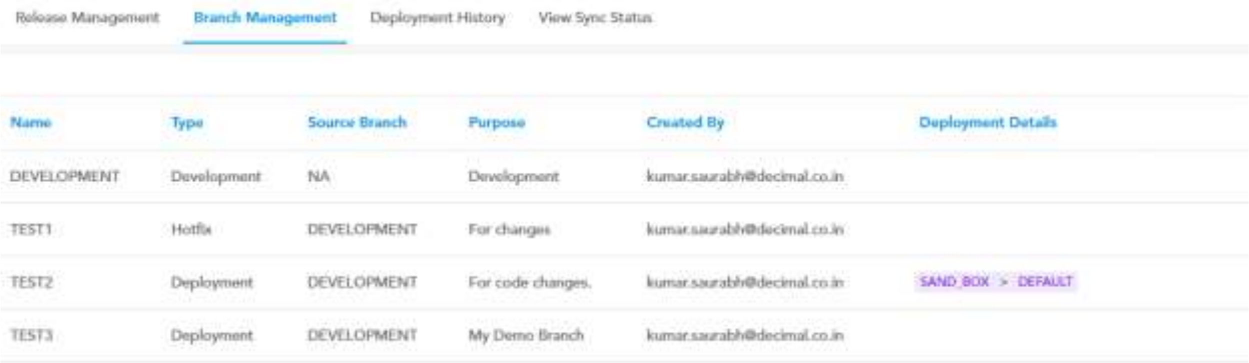
Name	Type	Source Branch	Purpose	Created By	Deployment Details
DEVELOPMENT	Development	NA	Development	kumar.saurabh@decimal.co.in	
TEST1	Hotfix	DEVELOPMENT	For changes	kumar.saurabh@decimal.co.in	
TEST2	Deployment	DEVELOPMENT	For code changes.	kumar.saurabh@decimal.co.in	SAND_B
TEST3	Deployment	DEVELOPMENT	My Demo Branch	kumar.saurabh@decimal.co.in	

The Test3 branch is a newly created deployment branch. It is created from the master branch.

3.2.2 Editing a Branch

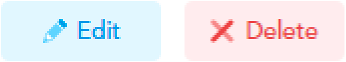
This feature allows you to edit the existing branch. To edit a branch:

1. In the **Release Manager** application, click the **Branch Management** tab, the application displays the list of the existing child branches.



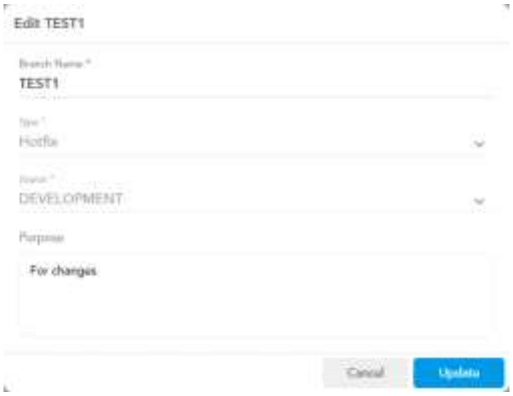
Name	Type	Source Branch	Purpose	Created By	Deployment Details
DEVELOPMENT	Development	NA	Development	kumar.saurabh@decimal.co.in	
TEST1	Hotfix	DEVELOPMENT	For changes	kumar.saurabh@decimal.co.in	
TEST2	Deployment	DEVELOPMENT	For code changes.	kumar.saurabh@decimal.co.in	SAND BOX > DEFAULT
TEST3	Deployment	DEVELOPMENT	My Demo Branch	kumar.saurabh@decimal.co.in	

2. Locate the name of the branch (For example: - Test1) that you want to edit.



3. Place the mouse pointer on the name of the branch, the **Edit** link appears.
4. Move the mouse pointer in the same row and then click **Edit**, the **Edit <Branch Name>** dialog box opens.
5. In the dialog box:

- ➢ Click in the **Branch Name** box and then change the name of the branch.
- ➢ Click in the **Purpose** box and then change the purpose of the branch (If required).


Edit TEST1

Branch Name *
TEST1

Type *
Hotfix

Source *
DEVELOPMENT

Purpose
For changes

Note:-
While editing a branch, the application allows you to edit the name and the purpose of the branch. The **Type** and the **Source** lists remain non-editable. It means that you cannot change the source branch from which the current branch is cut and also cannot change the type of the branch.

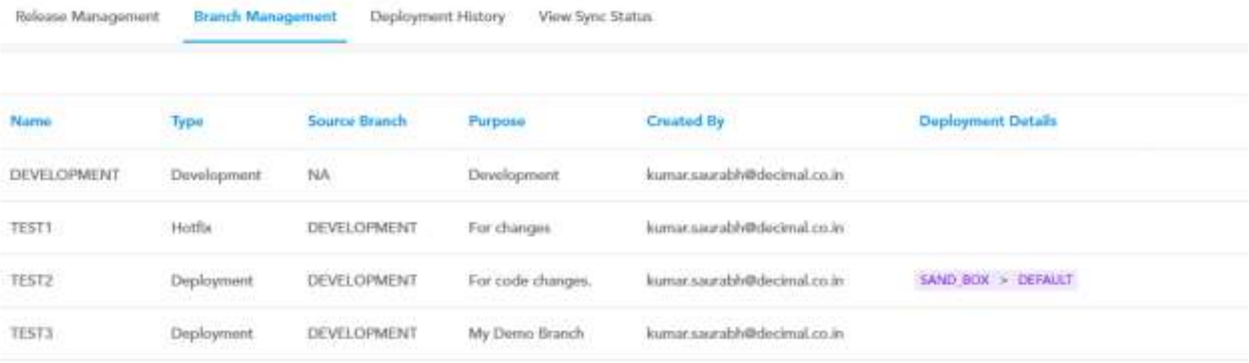
6. After you edit the name and the purpose of the branch, click **Update**, the branch is successfully edited.

3.2.3 Deleting a Branch

This feature allows you to delete a branch. You can only delete a non-master branch. The application does not allow you to delete the master branch. In the Release Manager application, a non-master branch can be a hot-fix, deployment, or backup branch that is created from the master branch.

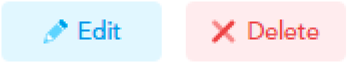
To delete a branch:

1. In the **Release Manager** application, click the **Branch Management** tab, the application displays the list of the existing child branches.



Name	Type	Source Branch	Purpose	Created By	Deployment Details
DEVELOPMENT	Development	NA	Development	kumar.saurabh@decimal.co.in	
TEST1	Hotfix	DEVELOPMENT	For changes	kumar.saurabh@decimal.co.in	
TEST2	Deployment	DEVELOPMENT	For code changes.	kumar.saurabh@decimal.co.in	SAND BOX > DEFAULT
TEST3	Deployment	DEVELOPMENT	My Demo Branch	kumar.saurabh@decimal.co.in	

2. In the list of branches, locate the name of the branch (For example: - Test3) that you want to delete.
3. Place the mouse pointer on the name of the branch, the **Delete** link appears.



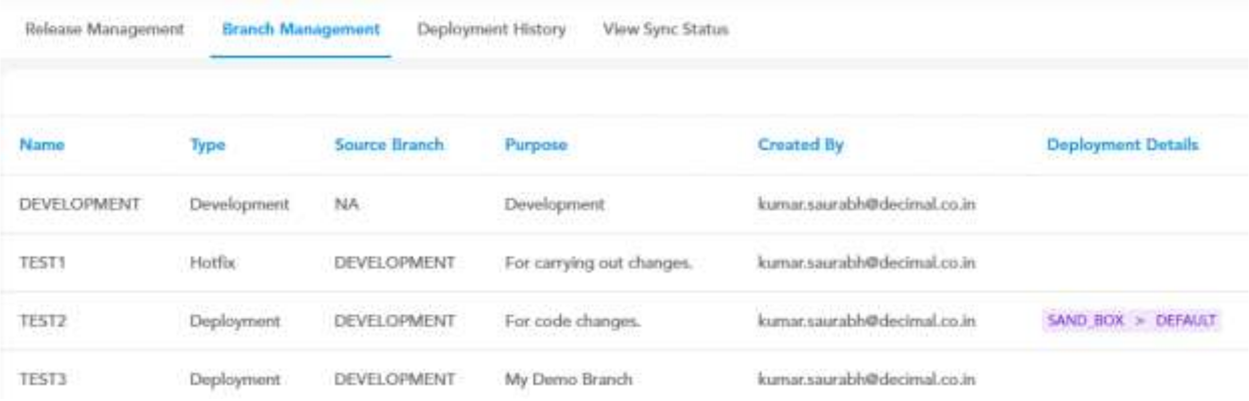
4. Move the mouse pointer in the same row and then click **Delete**, a message box opens and displays the message: Are you sure you want to delete this branch?
5. On the message box, click **OK**, the branch is deleted.

3.2.4 Viewing Branch Details

After you access the “Branch Management” module, the Release Manager application displays the details of the branches that have been created earlier. The Release Manager application displays the following details of a branch:

To view the details of the branch:

1. In the **Release Manager** application, click the **Branch Management** tab, the application displays the details of the earlier created branches.



Name	Type	Source Branch	Purpose	Created By	Deployment Details
DEVELOPMENT	Development	NA	Development	kumar.saurabh@decimal.co.in	
TEST1	Hotfix	DEVELOPMENT	For carrying out changes.	kumar.saurabh@decimal.co.in	
TEST2	Deployment	DEVELOPMENT	For code changes.	kumar.saurabh@decimal.co.in	SAND_BOX > DEFAULT
TEST3	Deployment	DEVELOPMENT	My Demo Branch	kumar.saurabh@decimal.co.in	

2. The details of the branch include:

Branch related Data	Description
Name	This field displays the name of the branch. The Release Manager consists of a default master branch, which is called the “Development” branch. When you create other non-master branches, you can define the name of the new branch.
Type	This field displays the following types of the branches: ➢ **Deployment** This type of the branch is created to fix or update the code of the application module. ➢ **Backup** This type of branch is created to take the backup of the application code. ➢ **Hotfix** This type of branch is created to fix the abruptly occurring errors in the application code.
Source Branch	This field displays the name of the branch from which the new branch is cut. You can cut the new branch from the master branch or from other non-master branch that you created earlier.
Purpose	This field displays the purpose of creating the new branch.
Created By	This field displays the name of the user who has created the branch.
Deployment Details	This field displays the deployment path that specifies the location where the application module/service is deployed. The deployment path contains the name of the environment where the application module is deployed.

3.3 Deployment History

This module allows you to access and view the details of the deployment. The deployment details include the followings:

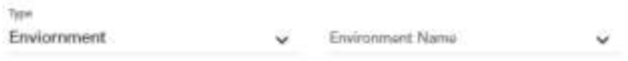
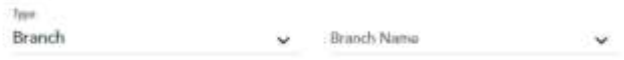
- **Name of the branch that is deployed**
- **Name of the environment where a branch is deployed**
- **The date and time at which a branch is deployed**
- **Registered email ID of the user who has deployed the branch**

Each time you deploy a branch into a specific environment, the Release Manager captures the deployment details and then helps you access them by using the “Deployment History” feature. This feature helps you access the deployment history on the basis of the two following criteria:

- **Name of environment**
- **Name of branch**

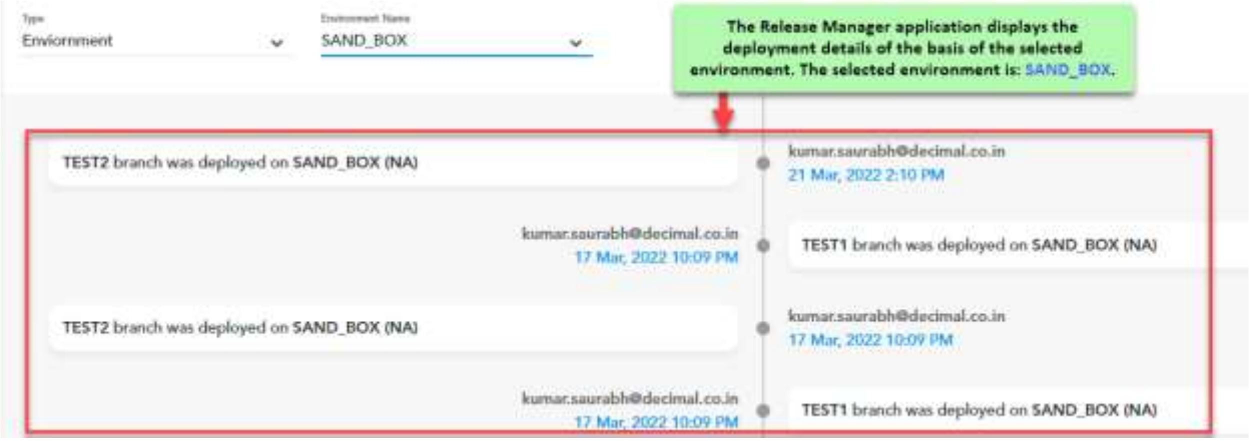
To access the deployment history:

1. In the **Release Manager** application, click the **Deployment History** tab, the Release Manager application displays the following lists:

List	Description
Type	After you click this list, it displays the following options: ➢ **Environment** Select this option to view the details of the deployment on the basis of a specific environment. ➢ **Branch** Select this option to view the details of the deployment on the basis of a specific branch.
Environment Name	The application displays this list if you select **Environment** in the **Type** list. In the **Environment Name** list, select an environment to view the deployment history on the basis of that selected environment.  Type Enviornment Environment Name
Branch Name	The application displays this list if you select **Branch** in the **Type** list. In the **Branch Name** list, select a branch to view the deployment history on the basis of that selected branch.  Type Branch Branch Name

2. After you select an environment or a branch, the Release Manager application displays the details of the deployment as follows:

- ➢ **Name of the environment**
- ➢ **Name of the branch**
- ➢ **The date and time at which a branch is deployed**
- ➢ **Registered email ID of the user who has deployed the branch**



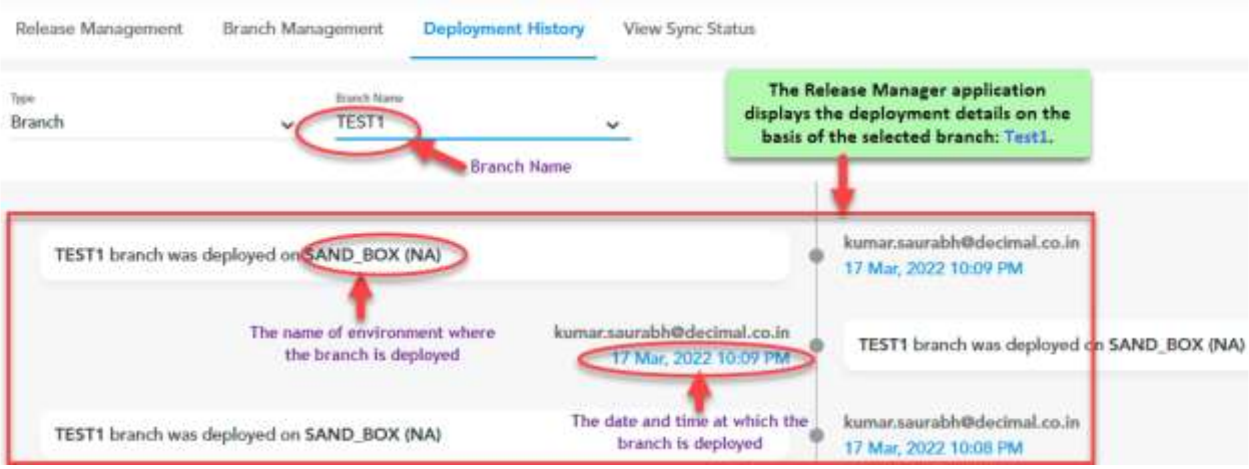
The Release Manager application displays the deployment details of the basis of the selected environment. The selected environment is: SAND_BOX.

TEST2 branch was deployed on SAND_BOX (NA)
kumar.saurabh@decimal.co.in
21 Mar, 2022 2:10 PM

TEST1 branch was deployed on SAND_BOX (NA)
kumar.saurabh@decimal.co.in
17 Mar, 2022 10:09 PM

TEST2 branch was deployed on SAND_BOX (NA)
kumar.saurabh@decimal.co.in
17 Mar, 2022 10:09 PM

TEST1 branch was deployed on SAND_BOX (NA)
kumar.saurabh@decimal.co.in
17 Mar, 2022 10:09 PM



The Release Manager application displays the deployment details on the basis of the selected branch: Test1.

TEST1 branch was deployed on SAND_BOX (NA)
kumar.saurabh@decimal.co.in
17 Mar, 2022 10:09 PM

The name of environment where the branch is deployed

TEST1 branch was deployed on SAND_BOX (NA)
kumar.saurabh@decimal.co.in
17 Mar, 2022 10:09 PM

The date and time at which the branch is deployed

TEST1 branch was deployed on SAND_BOX (NA)
kumar.saurabh@decimal.co.in
17 Mar, 2022 10:08 PM

3.4 Viewing Sync Status

This feature allows you to view the current status of the “Sync” activity for the specific service or application module. The “Sync” activity can have any of the two statuses: ***Pending*** or ***Success***. The “Pending” status means that the changes that you incorporate into the application module in the respective branch are yet to be synced on VRT. The “Success” status means that the changes that you incorporate into the application module have been successfully synced on VRT.

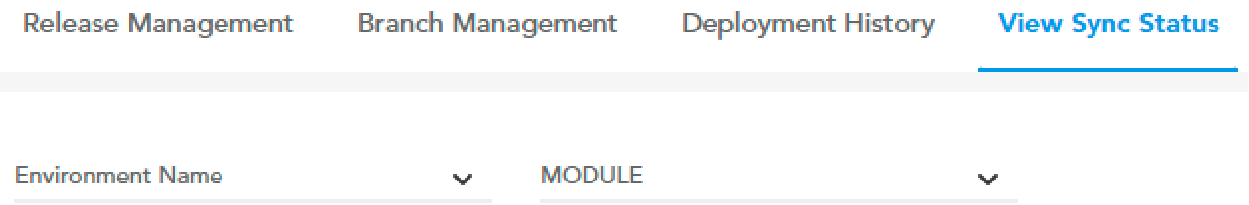
You can view the sync status by using the following selection criteria:

- **Environment Name**
- **Module Name**

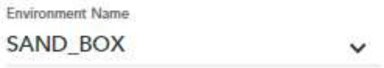
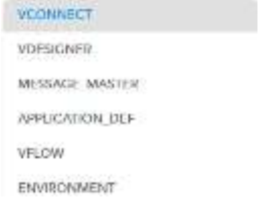
To view the sync status for the specific module:

1. In the **Release Manager** application, click the **View Sync Status** tab, the application displays the following lists:

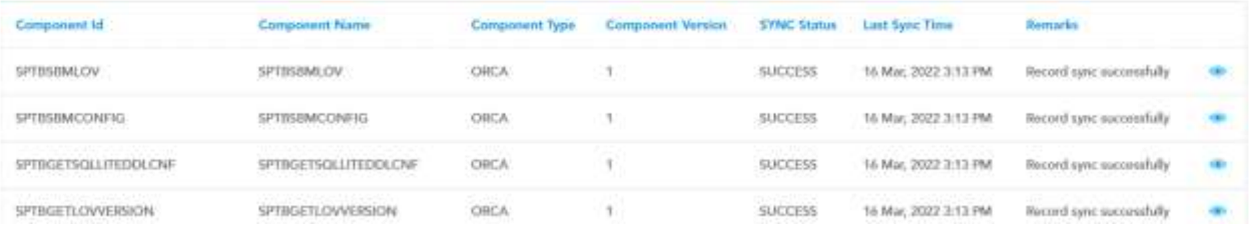
- **Environment Name**
- **Module**



2. In the lists, perform the following functions:

List	Description
Environment Name	Click this list and then select the environment in which you want to check the status of the “Sync” activity.  Environment Name SAND_BOX
Module	Click this list and then select the module for which you want to check the status of the “Sync” activity.  VCONNECT VDESIGNER MESSAGE_MASTER APPLICATION_DEF VFLOW ENVIRONMENT

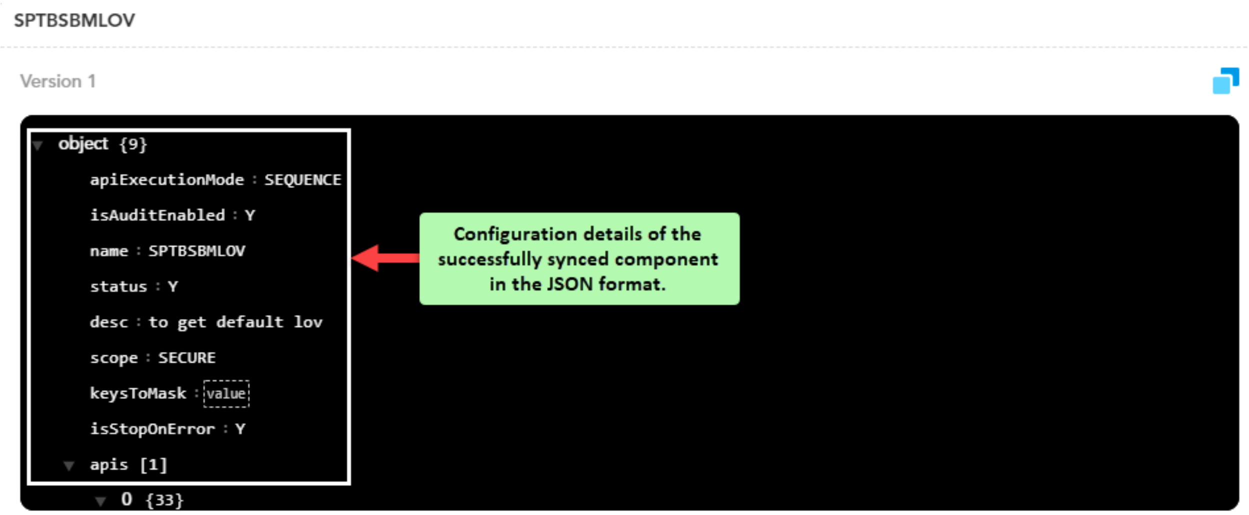
3. After you select the environment and the module, the application displays the sync details as follows:



Component Id	Component Name	Component Type	Component Version	SYNC Status	Last Sync Time	Remarks	
SPTBSBMLOV	SPTBSBMLOV	ORCA	1	SUCCESS	16 Mar, 2022 3:13 PM	Record sync successfully	👁
SPTBSBMCONFIG	SPTBSBMCONFIG	ORCA	1	SUCCESS	16 Mar, 2022 3:13 PM	Record sync successfully	👁
SPTBGETSQLLITEDDLCNF	SPTBGETSQLLITEDDLCNF	ORCA	1	SUCCESS	16 Mar, 2022 3:13 PM	Record sync successfully	👁
SPTBGETLOVVERSION	SPTBGETLOVVERSION	ORCA	1	SUCCESS	16 Mar, 2022 3:13 PM	Record sync successfully	👁

The sync details are described as below:

Field	Description
Component Id	This field displays the name of the component for which you cut a branch from the master branch. You access the respective component in the child branch, change or fix the application code, and then merge the branch into the master branch.
Component Name	This field displays the name of the component for which you cut a branch from the master branch.
Component Type	This field displays the type of the component. In the Release Manager application, the type of the component is defined at the database level. The Release Manager application maintains the type of the component as a unique name identifier.
Component Version	This field displays the updated version of the component after the child branch is merged into the master branch.
SYNC Status	The Release Manager application maintains two types of the sync status for the respective components: ➢ **Pending** The “Pending” status means that the changes that you incorporate into the application module in the respective branch are yet to be synced on VRT. ➢ **Success** The “Success” status means that the changes that you incorporate into the application module have been successfully synced on VRT.
Last Sync Time	This field displays the date and time in the ***dd Mon, yyyy hr:min PM/AM*** format only if the changes that are incorporated into the component have been successfully synced on VRT. This timestamp value (date and time) specifies the date and time at which the changes were successfully synced on VRT.
Remarks	This field displays the textual description related to the sync status.
Eye Icon (👁)	This field displays the clickable eye icon (👁). After you click the eye icon (👁), the application displays the configuration details of the successfully synced component in JSON format.



SPTBSBMLOV

Version 1

```
▼ object {9}
      apiExecutionMode : SEQUENCE
      isAuditEnabled : Y
      name : SPTBSBMLOV
      status : Y
      desc : to get default lov
      scope : SECURE
      keysToMask : value
      isStopOnError : Y
    ▼ apis [1]
        ▼ 0 {33}
```

Configuration details of the successfully synced component in the JSON format.

****